A sample implementation of CS349 A2.3 for Fall 2016 at University of Waterloo. This code is not for distribution. Implemented by Byron Weber Becker.

Implements a simple Entity-Relationship editor that supports the following features:

- Display entity relationships graphically.
- Display a list of entities.
- Display a list of their relationships.
- Add a new entity with the mouse.
- Select an entity and press “Delete” to delete it.
- Add a new relationship with the mouse.
- Rename an entity in the entity list.
- Rearrange entities in the graphic display.

Comments in the original code have been deleted before release to students. Present in the code reviewed in class.

EREdit

```
import model.ERModel;
import ui.*;
import javax.swing.*;
import java.awt.*;


/**
 * Created by bwbecker on 2016-09-30.
 */
public class EREdit {
    public static void main(String[] args) {

        ERModel model = new ERModel();
        TopLevelView view = new TopLevelView(model);

        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                JFrame f = new JFrame("EREdit");
                f.setSize(800, 500);
                f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                f.setContentPane(view);
                f.setVisible(true);
            }
        });
    }
}
```

ERModel

```
package model;

import java.awt.*;
import java.util.*;

/**
 * bwbecker 20161103
 */
public class ERModel extends Observable {

    public static final Object ADD_REMOVE_DATA = new Object();
    private Map<Integer, Entity> entities = new HashMap<Integer, Entity>();
    private Map<Integer, Arrow> arrows = new HashMap<Integer, Arrow>();
    private int nextUid = 100;

    public ERModel() {
        this.entities.put(4, new Entity(4, "Four", new Rectangle(30, 50, 100, 50)));
        this.entities.put(6, new Entity(6, "Six", new Rectangle(90, 200, 100, 50)));
        this.entities.put(8, new Entity(8, "Eight", new Rectangle(300, 200, 100, 50)));
        this.arrows.put(29, new Arrow(29, this.entities.get(4), this.entities.get(6)));
        this.arrows.put(30, new Arrow(30, this.entities.get(4), this.entities.get(8)));
    }

    public void addObserver(Observer o) {
        super.addObserver(o);
        o.update(this, this.ADD_REMOVE_DATA);
    }

    public Entity getEntity(int uid) {
        return this.entities.get(uid);
    }

    public void selectEntity(int uid) {
        Entity ent = this.entities.get(uid);
        for (Entity e : this.entities.values()) {
            e.setSelected(e.getUid() == uid);
        }

```

For the personal use of c4lang only. Not for redistribution.

```
71          for (Arrow a : this.arrows.values()) {
72              a.setSelected(a.fromEntity == ent);
73          }
74          this.setChanged();
75          this.notifyObservers();
76      }
77
78      public Entity getEntity(Point p) {
79          for (Entity e : this.entities.values()) {
80              if (e.contains(p)) {
81                  return e;
82              }
83          }
84          return null;
85      }
86
87      public Iterator<Entity> entityIterator() {
88          return this.entities.values().iterator();
89      }
90
91      public int[] getEntityUids() {
92          int[] uids = new int[this.entities.size()];
93
94          int i = 0;
95          for (Entity e : this.entities.values()) {
96              uids[i] = e.getUid();
97              i++;
98          }
99          Arrays.sort(uids);
100         return uids;
101     }
102
103     public void moveEntity(int uid, int dx, int dy) {
104         Entity e = this.getEntity(uid);
105         e.move(dx, dy);
106         this.setChanged();
107         this.notifyObservers();
108     }
109
110     public int addEntity(String name, Point topLeft, Point bottomRight) {
111         int uid = this.nextUid++;
112
113         this.entities.put(uid, new Entity(uid, name, new Rectangle(topLeft.x, topLeft.y,
114                 bottomRight.x - topLeft.x, bottomRight.y - topLeft.y)));
115
116         this.setChanged();
117         this.notifyObservers(this.ADD_REMOVE_DATA);
118         return uid;
119     }
120
121     public void setEntityName(int uid, String name) {
122         System.out.println("Setting entity name to " + name);
123         this.entities.get(uid).setName(name);
124         this.setChanged();
125         this.notifyObservers();
126     }
127
128     public void deleteSelectedEntity() {
129         Iterator<Map.Entry<Integer, Arrow>> aIter = this.arrows.entrySet().iterator();
130         while (aIter.hasNext()) {
131             Map.Entry<Integer, Arrow> entry = aIter.next();
132             if (entry.getValue().fromEntity.isSelected() ||
133                     entry.getValue().toEntity.isSelected()) {
134                 aIter.remove();
135             }
136         }
137
138         Iterator<Map.Entry<Integer, Entity>> eIter = this.entities.entrySet().iterator();
139         while (eIter.hasNext()) {
140             Map.Entry<Integer, Entity> entry = eIter.next();
141             if (entry.getValue().isSelected()) {
142                 eIter.remove();
143             }
144         }
145
146         this.setChanged();
147         this.notifyObservers(this.ADD_REMOVE_DATA);
148     }
149
150     public int[] getArrowUids() {
151         int[] uids = new int[this.arrows.size()];
152
153         int i = 0;
154         for (Arrow a : this.arrows.values()) {
155             uids[i] = a.getUid();
156             i++;
157         }
158         Arrays.sort(uids);
159         return uids;
160     }
161
162
```

For the personal use of c4lang only. Not for redistribution.

```
163     public Arrow getArrow(int uid) {
164         return this.arrows.get(uid);
165     }
166
167
168     public Arrow[] getArrows() {
169         Arrow[] arrows = new Arrow[this.arrows.size()];
170         this.arrows.values().toArray(arrows);
171         return arrows;
172     }
173
174     public Iterator<Arrow> arrowIterator() {
175         return this.arrows.values().iterator();
176     }
177
178
179     public void selectArrow(int uid) {
180         Arrow arrow = this.arrows.get(uid);
181
182         for (Entity e : this.entities.values()) {
183             e.setSelected(false);
184         }
185
186         for (Arrow a : this.arrows.values()) {
187             if (a.getUid() == uid) {
188                 a.fromEntity.setSelected(true);
189                 a.toEntity.setSelected(true);
190             }
191             a.setSelected(a.getUid() == uid);
192         }
193
194         this.setChanged();
195         this.notifyObservers();
196     }
197
198     public void addArrow(Entity from, Entity to) {
199         int uid = this.nextUid++;
200         this.arrows.put(uid, new Arrow(uid, from, to));
201         this.setChanged();
202         this.notifyObservers(this.ADD_REMOVE_DATA);
203     }
204
205 }
206
```

ERModelError

```
207
208 package model;
209
210 public class ERModelError extends Error {
211
212     public ERModelError(String msg) {
213         super(msg);
214     }
215
216     public ERModelError(String msg, Throwable cause) {
217         super(msg, cause);
218     }
219 }
220
```

DiagramPart

```
221
222 package model;
223
224 /**
225  * Created by bwbecker on 2016-11-03.
226  */
227 public class DiagramPart {
228
229     private int uid;
230     private boolean selected = false;
231
232     public DiagramPart(int uid) {
233         this.uid = uid;
234     }
235
236     public int getUid() {
237         return this.uid;
238     }
239
240     public boolean isSelected() {
241         return this.selected;
242     }
243
244     /* package */ void setSelected(boolean val) {
245         this.selected = val;
246     }
247 }
248
```

For the personal use of c4lang only. Not for redistribution.

Entity

```
package model;

import model.DiagramPart;
import java.awt.*;

/**
 * Created by bwbecker on 2016-11-03.
 */
public class Entity extends DiagramPart {

    private String name;
    private Rectangle location;

    public Entity(int uid, String name, Rectangle location) {
        super(uid);
        this.name = name;
        this.location = location;
    }

    public String getName() {
        return this.name;
    }

    public Rectangle getRect() {
        return (Rectangle)this.location.clone();
    }

    public boolean contains(Point p) {
        return this.location.contains(p);
    }

    /* package */ void setName(String name) {
        this.name = name;
    }

    /* package */ void move(int dx, int dy) {
        this.location.translate(dx, dy);
    }

    /* package */ Rectangle getLocation() {
        return this.location;
    }

}
```

Arrow

```
package model;

import java.awt.*;

/**
 * Created by bwbecker on 2016-11-03.
 */
public class Arrow extends DiagramPart {

    /* package */ Entity fromEntity;
    /* package */ Entity toEntity;

    public Arrow(int uid, Entity fromEntity, Entity toEntity) {
        super(uid);

        this.fromEntity = fromEntity;
        this.toEntity = toEntity;
    }

    public String fromName() {
        return this.fromEntity.getName();
    }

    public String toName() {
        return this.toEntity.getName();
    }

    public Point[] endPoints() {
        Rectangle from = this.fromEntity.getLocation();
        Rectangle to = this.toEntity.getLocation();
        int x1, y1, x2, y2;

        if (from.x + from.width < to.x) {
            x1 = from.x + from.width;
            x2 = to.x;
        } else if (from.x > to.x + to.width) {
            x1 = from.x;
            x2 = to.x + to.width;
        } else {
            x1 = from.x + from.width/2;
            x2 = to.x + to.width/2;
        }

        if (from.y + from.height < to.y) {
```

For the personal use of c4lang only. Not for redistribution.

```
            y1 = from.y + from.height;
            y2 = to.y;
        } else if (to.y + to.height < from.y) {
            y1 = from.y;
            y2 = to.y + to.height;
        } else {
            y1 = from.y + from.height/2;
            y2 = to.y + to.height/2;
        }

        return new Point[]{new Point(x1,y1), new Point(x2,y2)};
    }
}
```

TopLevelView

```
package ui;

import model.ERModel;
import javax.swing.*;
import java.awt.*;

/**
 * Created by bwbecker on 2016-11-03.
 */
public class TopLevelView extends JPanel {

    public TopLevelView(ERModel model) {
        ToolView toolView = new ToolView();
        GraphicView graphicView = new GraphicView(model, toolView);
        EntityView entityView = new EntityView(model);
        ArrowView arrowView = new ArrowView(model);

        JPanel lists = new JPanel(new GridLayout(2, 1));
        lists.add(entityView);
        lists.add(arrowView);

        JPanel left = new JPanel();
        left.setLayout(new BoxLayout(left, BoxLayout.Y_AXIS));
        left.add(toolView);
        left.add(Box.createVerticalStrut(30));
        left.add(lists);

        this.setLayout(new BorderLayout());
        this.add(graphicView, BorderLayout.CENTER);
        this.add(left, BorderLayout.WEST);

    }
}
```

ViewMode

```
package ui;

import java.util.Enumeration;

/**
 * Created by bwbecker on 2016-11-03.
 */
public enum ViewMode {
    NORMAL,
    ADD_ENTITY,
    ADD_ARROW,
    MOVE_ENTITY
}
```

ToolView

```
package ui;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import static ui.ViewMode.*;

/**
 * Created by bwbecker on 2016-11-03.
 */
public class ToolView extends JPanel {

    private JButton addEntity = new JButton("Add Entity");
    private JButton addArrow = new JButton("Add Arrow");

    ViewMode viewMode = NORMAL;
```

For the personal use of c4lang only. Not for redistribution.

```
    public ToolView() {
        super();
        this.setLayout(new BoxLayout(this, BoxLayout.X_AXIS));

        this.add(this.addEntity);
        this.add(this.addArrow);

        this.addEntity.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                viewMode = ADD_ENTITY;
            }
        });

        this.addArrow.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                viewMode = ADD_ARROW;
            }
        });

    }

    public ViewMode getViewMode() {
        return this.viewMode;
    }
}
```

GraphicView

```
package ui;

import model.Arrow;
import model.ERModel;
import model.Entity;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Iterator;
import java.util.Observable;
import java.util.Observer;

import static ui.ViewMode.*;


/**
 * Created by bwbecker on 2016-11-02.
 */
public class GraphicView extends JComponent implements Observer {

    private ERModel model;
    private Point mouseDownPoint = null;
    private Point currPoint = null;
    private Entity selectedEntity = null;
    private ToolView toolView;


    public GraphicView(ERModel model, ToolView toolView) {
        this.model = model;
        this.toolView = toolView;

        this.addMouseListener(new MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent e) {
                super.mousePressed(e);
                switch (toolView.getViewMode()) {
                    case NORMAL:
                        mouseDownPoint = e.getPoint();
                        selectedEntity = model.getEntity(mouseDownPoint);
                        break;

                    case ADD_ENTITY:
                    case ADD_ARROW:
                        mouseDownPoint = e.getPoint();
                        break;
                }
            }

            @Override
            public void mouseReleased(MouseEvent e) {
                super.mouseReleased(e);
                switch (toolView.getViewMode()) {
                    case NORMAL:
                        Point p = e.getPoint();
                        Entity ent = model.getEntity(p);
                        if (ent != null) {
                            setFocusable(true);
                            requestFocusInWindow();
```

For the personal use of c4lang only. Not for redistribution.

```
                model.selectEntity(ent.getUid());
            }
            break;

        case ADD_ENTITY:
            int uid = model.addEntity("", mouseDownPoint, e.getPoint());
            JTextField name = new JTextField(15);
            name.setLocation(mouseDownPoint.x, mouseDownPoint.y);
            name.setSize(e.getPoint().x - mouseDownPoint.x, 25);
            name.addFocusListener(new FocusAdapter() {
                @Override
                public void focusLost(FocusEvent fe) {
                    model.setEntityName(uid, name.getText());
                    remove(name);
                    repaint();
                }
            });
            name.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent ae) {
                    model.setEntityName(uid, name.getText());
                    remove(name);
                    repaint();
                }
            });
            add(name);
            name.requestFocus();
            repaint();

            break;

        case ADD_ARROW:
            Entity endEntity = model.getEntity(e.getPoint());
            Entity startEntity = model.getEntity(mouseDownPoint);
            if (endEntity != null && startEntity != null) {
                model.addArrow(startEntity, endEntity);
            } else {
                repaint();
            }
            break;
        }
        selectedEntity = null;
        toolView.viewMode = NORMAL;
    }
});
```

```
    this.addMouseMotionListener(new MouseMotionAdapter() {
        @Override
        public void mouseDragged(MouseEvent e) {
            super.mouseDragged(e);
            switch (toolView.getViewMode()) {
                case NORMAL:
                    if (selectedEntity != null) {
                        Point p = e.getPoint();
                        int dX = p.x - mouseDownPoint.x;
                        int dY = p.y - mouseDownPoint.y;
                        model.moveEntity(selectedEntity.getUid(), dX, dY);
                        mouseDownPoint = p;
                    }
                    break;

                case ADD_ENTITY:
                case ADD_ARROW:
                    currPoint = e.getPoint();
                    repaint();
                    break;
            }
        }
    });

    this.addKeyListener(new KeyAdapter() {
        @Override
        public void keyPressed(KeyEvent e) {
            if (e.getKeyCode() == KeyEvent.VK_DELETE) {
                model.deleteSelectedEntity();
            }
        }
    });

    this.model.addObserver(this);
}


public void update(Observable obs, Object obj) {
    this.repaint();
}


private Stroke regular = new BasicStroke(1.0F);
private Stroke hilite = new BasicStroke(2.0F);
private Stroke dashed = new BasicStroke(1, BasicStroke.CAP_BUTT,
BasicStroke.JOIN_BEVEL, 0, new float[]{9}, 0);
```

For the personal use of c4lang only. Not for redistribution.

```
603
604     public void paintComponent(Graphics g) {
605         super.paintComponent(g);
606         Graphics2D g2 = (Graphics2D) g;
607
608         g2.drawRect(0, 0, this.getWidth() - 1, this.getHeight() - 1);
609
610         Iterator<Arrow> ait = this.model.arrowIterator();
611         while (ait.hasNext()) {
612             Arrow a = ait.next();
613             Point[] ep = a.endPoints();
614             if (a.isSelected()) {
615                 g2.setStroke(hilite);
616             } else {
617                 g2.setStroke(regular);
618             }
619             g2.drawLine(ep[0].x, ep[0].y, ep[1].x, ep[1].y);
620             g2.drawOval(ep[1].x - 4, ep[1].y - 4, 8, 8);
621         }
622
623         Iterator<Entity> eit = this.model.entityIterator();
624
625         while (eit.hasNext()){
626             Entity e = eit.next();
627             Rectangle r = e.getRect();
628             g2.setColor(Color.WHITE);
629             g2.fillRoundRect(r.x, r.y, r.width, r.height, 10, 10);
630
631             g2.setColor(Color.BLACK);
632             if (e.isSelected()) {
633                 g2.setStroke(hilite);
634                 g2.drawRoundRect(r.x, r.y, r.width, r.height, 10, 10);
635                 g2.setStroke(regular);
636             } else {
637                 g2.drawRoundRect(r.x, r.y, r.width, r.height, 10, 10);
638             }
639             g2.drawString(e.getName(), r.x + 4, r.y + r.height / 2);
640         }
641
642         if (this.toolView.getViewMode() == ADD_ENTITY) {
643             Point pp = this.mouseDownPoint;
644             Point cp = this.currPoint;
645             if (pp != null && cp != null) {
646                 g2.setStroke(dashed);
647                 g2.drawRoundRect(pp.x, pp.y, cp.x - pp.x, cp.y - pp.y, 10, 10);
648             }
649         } else if (this.toolView.getViewMode() == ADD_ARROW) {
650             Point pp = this.mouseDownPoint;
651             Point cp = this.currPoint;
652             if (pp != null && model.getEntity(pp) != null) {
653                 g2.setStroke(dashed);
654                 g2.drawLine(pp.x, pp.y, cp.x, cp.y);
655                 g2.drawOval(cp.x - 4, cp.y-4, 8, 8);
656             }
657         }
658     }
659 }
```

EntityView

```
660 package ui;
661
662 import model.ERModel;
663 import model.Entity;
664 import javax.swing.*;
665 import javax.swing.event.ListSelectionEvent;
666 import javax.swing.event.ListSelectionListener;
667 import javax.swing.table.AbstractTableModel;
668 import java.awt.*;
669 import java.util.Observable;
670 import java.util.Observer;
671
672 /**
673  * Created by bwbecker on 2016-11-03.
674  */
675 public class EntityView extends JPanel implements Observer {
676
677     private ERModel model;
678     private JLabel label = new JLabel("Entities");
679     private EntityTable eTable = new EntityTable();
680     private JTable table = new JTable(eTable);
681
682     private boolean updatingSelections = false;
683
684     public EntityView(ERModel model) {
685         this.model = model;
686
687         table.setPreferredScrollableViewportSize(new Dimension(200, 70));
688         table.setFillsViewportHeight(true);
689
690
691
```

For the personal use of c4lang only. Not for redistribution.

```
692         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
693         this.add(this.label);
694         this.add(new JScrollPane(this.table));
695 
696 
697         this.table.getSelectionModel().addListSelectionListener(new
698 ListSelectionListener() {
699             @Override
700             public void valueChanged(ListSelectionEvent e) {
701                 if (!updatingSelections) {
702                     if (!e.getValueIsAdjusting()) {
703                         model.selectEntity(eTable.uids[table.getSelectedRow()]);
704                     }
705                 }
706             }
707         });
708 
709         this.model.addObserver(this);
710     }
711 
712     public void update(Observable obs, Object obj) {
713         this.eTable.refreshUids();
714 
715         ListSelectionModel lsm = this.table.getSelectionModel();
716         this.updatingSelections = true;
717         lsm.clearSelection();
718         for (int i = 0; i < this.eTable.uids.length; i++) {
719             if (this.model.getEntity(this.eTable.uids[i]).isSelected()) {
720                 lsm.addSelectionInterval(i, i);
721             }
722         }
723         this.updatingSelections = false;
724     }
725 
726     private class EntityTable extends AbstractTableModel {
727 
728         int[] uids = new int[0];
729 
730         void refreshUids() {
731             this.uids = model.getEntityUids();
732             updatingSelections = true;
733             this.fireTableDataChanged();
734             updatingSelections = false;
735         }
736 
737         @Override
738         public int getRowCount() {
739             return uids.length;
740         }
741 
742         @Override
743         public String getColumnName(int colIndex) {
744             switch (colIndex) {
745                 case 0:
746                     return "UID";
747                 case 1:
748                     return "Entity Name";
749                 default:
750                     return "";
751             }
752         }
753 
754         @Override
755         public boolean isCellEditable(int row, int col) {
756             return col == 1;
757         }
758 
759         @Override
760         public int getColumnCount() {
761             return 2;
762         }
763 
764         @Override
765         public Object getValueAt(int rowIndex, int columnIndex) {
766             Entity ent = model.getEntity(this.uids[rowIndex]);
767             assert (ent != null);
768 
769             switch (columnIndex) {
770                 case 0:
771                     return ent.getUid();
772                 case 1:
773                     return ent.getName();
774             }
775             throw new Error("Fell through switch with columnIndex = " +
776 columnIndex);
777         }
778 
779         @Override
780         public void setValueAt(Object val, int rowIndex, int colIndex) {
781 
782             if (colIndex == 1) {
783                 model.setEntityName(this.uids[rowIndex], val.toString());
```

For the personal use of c4lang only. Not for redistribution.

```
            }
        }
    }
}
```

ArrowView

```
package ui;

import model.Arrow;
import model.ERModel;
import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.AbstractTableModel;
import java.awt.*;
import java.util.Observable;
import java.util.Observer;

/**
 * Created by bwbecker on 2016-11-02.
 */
public class ArrowView extends JPanel implements Observer {

    private ERModel model;
    private JLabel label = new JLabel("Arrows");
    private ArrowTable aTable = new ArrowTable();
    private JTable table = new JTable(aTable);

    private boolean updatingSelections = false;

    public ArrowView(ERModel model) {
        this.model = model;

        table.setPreferredScrollableViewportSize(new Dimension(200, 70));
        table.setFillsViewportHeight(true);


        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
        this.add(this.label);
        this.add(new JScrollPane(this.table));


        this.table.getSelectionModel().addListSelectionListener(new
ListSelectionListener() {
            @Override
            public void valueChanged(ListSelectionEvent e) {
                // See note on updatingSelections
                if (!updatingSelections) {
                    if (!e.getValueIsAdjusting()) {
                        model.selectArrow(aTable.uids[table.getSelectedRow()]);
                    }
                }
            }
        });

        this.model.addObserver(this);
    }

    public void update(Observable obs, Object obj) {
        if (obj == (model.ADD_REMOVE_DATA)) {
            this.aTable.refreshUids();
        }

        ListSelectionModel lsm = this.table.getSelectionModel();
        this.updatingSelections = true;
        lsm.clearSelection();
        for(int i=0; i<this.aTable.uids.length; i++) {
            if (this.model.getArrow(this.aTable.uids[i]).isSelected()) {
                lsm.addSelectionInterval(i, i);
            }
        }
        this.updatingSelections = false;
    }


    private class ArrowTable extends AbstractTableModel {

        private int[] uids = new int[0];

        void refreshUids() {
            this.uids = model.getArrowUids();
            updatingSelections = true;
            this.fireTableDataChanged();
            updatingSelections = false;
        }

        @Override
```

For the personal use of c4lang only. Not for redistribution.

```
873         public int getRowCount() {
874             return uids.length;
875         }
876 
877         @Override
878         public int getColumnCount() {
879             return 3;
880         }
881 
882         @Override
883         public String getColumnName(int colIndex) {
884             switch (colIndex) {
885                 case 0: return "UID";
886                 case 1: return "From Entity";
887                 case 2: return "To Entity";
888                 default: return "";
889             }
890         }
891 
892         @Override
893         public Object getValueAt(int rowIndex, int columnIndex) {
894             Arrow arrow = model.getArrow(this.uids[rowIndex]);
895             assert(arrow != null);
896 
897             switch (columnIndex) {
898                 case 0:
899                     return arrow.getUid();
900                 case 1:
901                     return arrow.fromName();
902                 case 2:
903                     return arrow.toName();
904             }
905             throw new Error("Fell through case with columnIndex = " + columnIndex);
906         }
907     }
908 
909 }
910 
```

For the personal use of c4lang only. Not for redistribution.