# Concept Explorer. The User Guide

September 12, 2006

## Introduction

### What is it?

This is a version 1.3 of "Concept Explorer"(ConExp) tool, that implements basic functionality needed for study and research of Formal Concept Analysis(FCA).

Formal Concept Analysis is a branch of lattice theory, that was developed starting from early 1980-ies by members of Rudolf Wille's group in Darmstadt.

It can be used for analysis of simple attribute object tables (called context in FCA) and exploration of different dependencies, that exists between attributes.

For more information about Formal Concept Analysis, see http://www.math.tu-dresden.de/~ganter/fba.html and http://www.fcahome.org.uk/

ConExp is released under BSD-style license. Please read file license.txt in the distribution. You can read license files of libraries, that are used in ConExp, reading file, containing "license" in its name and first name of which corresponds to the name of the library jar file.

If you use ConExp in scientific research, please cite the following article [1].

### What can I do with it?

ConExp provides the following functionality:

- context editing
- building concept lattices from context
- finding bases of implications that are true in context
- finding bases of association rules that are true in context
- performing attribute exploration

### A little bit of history

ConExp was first developed as a part of master's thesis under the supervision of Prof. Dr. Tatyana Taran at the National Technical University of Ukraine "KPI" in 2000. During the following years, it was extended and now is an open source project on Sourceforge [5].

# ConEmp installation

## Required software

In order to run ConExp, the Java Runtime Environment version 1.4 or higher is required. It is usually recommended to use latest version of JRE (1.5.0 on the moment of writing). If you don't have it, you can get them from the following URL: http://java.sun.com/j2se/downloads.html

## Installation process

Just unpack the content of zip or tar.gz file into preferred location. Make sure, that java can be called from this location (Just open the console in selected directory and type java -version in order to check availability of java on your system. If java is installed, then information about environment and version of java software should be displayed)

# Working with Concept Explorer

## Starting the work

Run the appropriate start script("conexp.bat" on Windows, "conexp.sh" on Unix). On Unix: before that set the executable attribute for "conexp.sh". If you run it from the command line, make sure, that you are in the installation directory.

Alternatively, on all platforms you can run "java -jar conexp.jar" from the command line. On Windows use "javaw -jar conexp.jar" if you don't want to see the java console.

## Concept Explorer user interface overview

ConExp user interface consists of the following parts:

### Menu

**Main toolbar** - Contains buttons for global application operations - "Create new document", "Open", "Save", "Compute the number of concepts", "Compute concept lattice", "Perform attribute exploration", "Calculate the Duquenne-Guigues set of implications", "Calculate association rules" and also a combo box, that allows to select the update mode for document components that are computed from context - lattices, implication and association rule sets. Currently, the two update mode are supported - clearing of affected component or recomputation of affected components. The first mode is recommended, when user is going to make a lot of changes in context, or the size of context is big. In this mode affected components are cleared and can be recomputed by calling compute operation for the corresponding view type. In the second mode, the affected components are recomputed after each change of context. This may lead to big computational expenses if context is big or dense.

**Main pane.** Main pane includes document tree, option pane, and view pane.

**Document tree** displays the structure of the current document and allows to navigate between different views (i.e. context view, lattice visualization view(s), implications view and association rules view).

**Option pane** allows to edit different options connected with view.

**View pane** contains the display for current view. For each view there is a corresponding toolbar with view-specific operations .

**Status bar**

## Creating the new document

Usually, the new document is created on the start of work with ConExp. Alternatively, one can create a new document by pressing "New context" button on main toolbar or selecting "New" menu item in "Files" menu.

If there was a previously opened and modified document, user will be prompted to save it (or cancel creating new document) before the creation of a new document.

### Opening existing documents

ConExp allows to work with several different data formats. It is possible to work with contexts, that were created using ConImp [3] .

Currently the following formats are supported:

**cex** - ConExp native format. This is XML-based format. Stores information about context and lattice line diagram, and also, whether implications and/or associative rules were calculated. It is recommended to use this format for work.

**cxt** - ConImp context data. Only context can be stored in such format.

**csv** - Comma - Separated values - For now, only import of context is supported for this format. Actual separator is semicolon(;). It is assumed, that the first line of file contains attributes names, and first cell is empty (I.e, if one has a context with attributes attr1 and attr2, then first line will be the following:";attr1;attr2". Each next line should start from object name and then sequence of 0 and 1. In cells, in which 1 is set, cross will be put in imported context.

**oal** - Object Attribute list - For now, only import of context is supported for this format. Each line presents information about object and then which attributes this object possess. If object obj1 has attributes attr2 and attr3, then line for obj1 will look as follows: "obj1:attr2;attr3"

Also it is possible to reopen document, on which you were working before, by selecting one of the items in “Reopen” sub-menu in “Files” menu.

## Saving your work

In order to save your work, use menu items “Save” and “Save as” in “Files” menu, or “Save file” button on main toolbar. The recommended storage format is native ConExp format

## Working with context

### Undo/redo support

For all operations, that are performed on context, undo/redo support is provided. One can undo the performed operations by pressing “Undo last action” button and redo by pressing “Redo last action” button on context editor’s toolbar.

### Changing size of context

In order to change the size of context, one should use the properties window on the left side of window, and enter new number of objects/attributes in “object count”/”attribute count” properties.

Also it is possible to add new object(attribute) into context by pressing “Add object”/”Add attribute” button on context editor’s toolbar.

In order to remove some set of object/attributes, select them in context editor and then perform “Remove object(s)”/”Remove attribute(s)” action from context editor’s context menu.

### Compressed view of context

If one want to get a better overview over the context, one can select “Compressed” option on context editor’s property pane. Then the width of context’s columns will be set just to fit value of cross, and one can have a better look on structure of context.

### Arrow relation visualization

In order to visualize arrow relation, select “show arrow” in property “Show arrow relations”. (If you don’t know, what it is, probably, you don’t need it. To learn more about arrow relations, have a look at book [2])

### Entering data into the context

**Fast editing of contexts** If one need to input a context of moderate size, one can use the so-called fast context editing.

Just use keys “x” and “.” , when staying in cell in relation area, than the cross or blank value will be entered in current cell and cursor will move to the next cell in relation area.

**Transformations on selected areas** After selecting the area of cell, one can perform transformations of content of incidence relation between objects and attributes.

The following transformation are supported:

**Fill selection** - fill the selected area of incidence relation with crosses.

**Clear selection** - clears the content of selected area of incidence relation

**Inverse selection** - replaces in selected area of incidence relation crosses with one and vice-versa.

All these transformations can be performed by using appropriate command from context menu.

### Operations on contexts

Following operations can be performed on contexts:

**object clarification** - replacing in context objects, that have equal sets of attributes, with one (first from beginning of context) object. This operation is invoked by pressing “Clarify objects” button on context editor’s toolbar.

**attribute clarification** - analogous operation, but only on attribute set. Is invoked by pressing “Clarify attributes” button on context editor toolbar.

**object set reduction** - removing from object set all objects, that can be obtained as a result of intersection of some other objects. In process of performing reduction clarification is also performed. This operation doesn’t change the structure of the concept lattice - concept lattice of reduced context is isomorphic to the concept lattice of the original context. Operation is performed by pressing “Reduce objects” button on context editor’s toolbar.

**attribute set reduction** - analogous operation on attribute set. Operation is performed by pressing “Reduce attributes” button on context editor’s toolbar.

**context reduction** - simultaneous application of object and attribute set reduction. Performed by pressing “Reduce context” button on context editor’s toolbar.

**transposition** - exchange of objects and attribute set and corresponding change of relation between them. Performed by pressing “Transpose context” button on context editor’s toolbar.

## Exploring the line diagram

### Building the concept lattice

In order to build concept lattice, use button “Build Lattice” on the main toolbar. After some time, that depends from complexity of lattice, the drawing of lattice (also named **line diagram**) will appear. (Remark - layout of lattice is time consuming operation, that’s why first only drawing, consisting from one node can appear, and then again, after some time, the layouted lattice will appear).

### Interpreting the drawing

Concept lattice represents the univocal transformation of the context.

Top element of the lattice corresponds to the unit element of concept lattice. Bottom element of concept lattice represents the zero element of concept lattice. Each node of lattice corresponds to so called **(formal) concept** in Formal Concept Analysis - a pair (O, A), where O - set of objects and A - set of attributes, such, that A contains all attributes, that all objects from O have in common and only this attributes, and O contains all objects from context, that has set attributes A among their attributes. Set of objects O is called **extent** of concept (O, A) and set of attributes A is called **intent** of set of attributes A.

So called reduced labeling is used in order to succinctly represent information about intents and extents of formal context. If label of attribute A is attached to some concept, that means, that this attribute occurs in intents of all concepts, reachable by descending paths from this concept to zero concept (bottom element) of lattice. If label of object O is attached to some concept, this means, that object O lays in extents of all concepts, reachable by ascending paths in lattice graph from this concept to unit concept (top element) of lattice.

If drawing of node contains blue filled upper semicircle, that means, that there is an attribute, attached to this concept. If drawing of node contains black filled lower semicircle, that means, that there is a object, attached to this concept.

Sometimes node or edge in line diagrams is displayed in red color. This means, that this edge or node are located very near or overlap with some other node. In order to improve layout, try manual adjustment of layout or some other layout.

### Visualization modes

Basically, there is two visualization modes, that behave differently, when drawing of lattice doesn’t fit into the existing screen estate.

They are:

**scrolling mode** - when drawing of lattice doesn’t fit into the screen estate, that virtual window is enlarged and user can see only some part of lattice drawing. This mode is activated by default.

**fit to screen mode** - drawing of lattice is rescaled in order to fit into the available screen estate.

Switching between scrolling and fit to screen mode is performed with the help of "Scale picture to fit into the image" button on lattice visualization pane toolbar. Pressing this button toggles between first mode and second and vice versa.

The following commands make sense only in scrolling mode:

**Grab and drag** - this command performs panning of the visible area. After pressing this button, the cursor changes to cross and user can pan the drawing. To switch off this mode, press "Grab and Drag" button one more time.

**Zoom in, Zoom out, No zoom** - these commands perform actions, corresponding to their names.

### Changing visualization options

The following visualization options can be adjusted via in "drawing options" properties pane on left part of the screen:

**Attribs** - upper label visualization mode. Possible values are "Show labels" - show attribute's label at corresponding concept (See also remark before about reduced labeling).

**Objects** - lower label visualization mode. Possible values are

- **Don't show** - no labels are shown
- **Show labels** - show object labels below the corresponding concepts
- **Show own objects** - for concepts, that has some objects attached (has non empty object **contingen**t) show number and percentage of objects, that belong exactly (i.e., their attribute set is equal to intent) to this concept
- **Show object count** - for every node show, exactly what number (percentage) of object lay in extent of this node's concept.
- **Stability** - for every node show, what minimal number of objects should be removed from context, that node with such intent disappeared from concept lattice.

**Draw node** - this option specify, how the radius of node is calculated. The possible values are:

- **~to own objects** - node radius is calculated proportionally to size of contingent (amount of objects, that match intent of this node exactly)
- **fixed radius** - all nodes has equal node radius. Actual node radius is determined by option "Node radius"

**~of object extent** - node radius is calculated proportionally to size of its extent.

**stability** - node radius is calculated proportionally to its stability to destruction (see description of Stability above).

**Draw edge** - specifies, how exactly edge is drawn. The possible values are:

**one pixel** - edge width is fixed

**no** - edge is not drawn

**~object** - proportionally to the number of objects, that "pass" through this edge. Equivalent of "~of object extent" option for drawing node

**~connection** - edge size is proportional to ratio between extent size of lower and upper concept, that are connected by edge. This value is equal to confidence of approximate association rule, that corresponds to edge.

**Highlight** - specifies, which nodes are highlighted, except for selected edges. These options were created in order to make exploration of lattice easier. Possible values of this option are:

**Filter and ideal** - nodes of filter (all nodes, that are reachable by ascending paths from selected node to top of lattice) and ideal ( all nodes, that are reachable by descending paths from selected node to bottom of lattices) are highlighted

**Selected** - only selected node is highlighted

**Neighbors** - selected node and it's upper and lower neighbors are highlighted

**Ideal** - nodes of ideal are highlighted

**Filter** - nodes of filter are highlighted

**No** - no nodes are highlighted. This option may be useful for storing images of lattice.

**Label font size** - specifies the size of font, that is used for upper and lower labels.

**Grid size x** - specifies the preferred distance by x coordinate between different nodes on one level of drawing. It is used as parameter for layout and after layout change of this value leads to rescaling of coordinates of nodes by x scale

**Grid size y** -specifies the preferred distance by y coordinate between nodes on adjacent levels of drawing. It is used as parameter for layout and after layout change of this value leads to rescaling of coordinates of nodes by y scale.

**Node radius** - this parameter specifies the maximal possible radius of concept node and is used when drawing nodes.

### Changing layout of lattice

If the initial drawing of lattice is not very satisfactory, than it is recommended to try to perform several different layouts in order to find the best first approximation before starting to perform manual adjustment of drawing.

**Warning:** performing layout of lattice is irreversible operation (for now). Don't do it, if you have done adjustments, that you would not like to loose.

Several algorithms have different options, that can be access through "Layout options" tab in properties panel.

The following layout algorithms are provided:

**Minimal intersections** - this is adapted to lattices version of algorithm for laying out hierarchical graphs. It tries to minimize number of intersections between edges. It has no parameters. Usually this algorithm provides best results, but it is pretty slow for the big lattices.

**Chain decomposition** - adapted version of algorithms of chain decomposition by M. Skorsky. This algorithms builds so called **additive lines diagrams.** It's recommended to use ideal node movement strategy, when working with such line diagrams. This algorithm produces very good results for distributive lattices. Chain decomposition algorithms has following options:

> **Representation** - what kind of representation is used for concept, when his coordinates are calculated. Can be either attribute-based or object-based.
>
> **Placement** - determines assignment of values to set of vectors. Can take one of three possible values - *exponential*, *straight* or *angular*.
>
> **Rotate left, Rotate right** - performs rotation of set of attribute vectors - is used to select the best one from several possible.

The next algorithms belongs to family of so-called "force-directed" layout algorithms. They are:

**Freese layout** - adaptation of Ralph Freese [4] algorithm for drawing lattices. Algorithms has following parameters:

> **Attraction** - regulates the attraction force between nodes
>
> **Repulsion** - regulates the repulsion force between nodes.
>
> **Angle** - this is not actually a parameter of the algorithms. Freese algorithms performs layout in 3D space, and angle parameter controls the angle, which is used for projecting results of layout in 3D space of 2D surface of screen

**Force layout** - other force directed algorithm, that differs from previous one by the way, how the forces are calculated. Parameters of this algorithm are analogous to the parameters of the previous one.

### Manually adjustment of the drawing

Unfortunately, for now no one layout algorithms, that produce good results for all types of lattice is known. So, the last way to produce good drawing is to perform the manual adjustment of the lattice.

Movement of lattice nodes is constrained in ConExp in order to maintain a correct parent-child (successor-predecessor) relations between nodes.

Following tools exists in ConExp in order to help manually adjust a lattice:

**Ideal node movement mode** - when moving a node, the whole ideal of the node is moved. The switch between this mode and one node movement strategy is performed by pressing "Toggle node move mode" button.

**Align nodes to grid** - performs alignment of node coordinates to the invisible grid of size 8 on 8 pixels.

### Storing default lattice drawing settings

As there are a lot of options for performing layout of line diagram, it is possible for user to store some set of settings as default. In order to do so, press "Store preferences as default" button on lattice visualization pane toolbar. After that, these preferences will be used as default for all newly computed concept lattices.

**Storing images of drawing** One of the most frequent uses of ConExp is to produce images of lattice drawings for some future usage. This task can be achieved by creating a good drawing of lattice and then pressing the "Save lattice image" button on lattice visualization pane toolbar. Currently saving image to png and jpeg is supported.

### Building lattices of subcontexts

ConExp also provides the ability to build lattices, that corresponds to subcontext of the original context. This task can be achieved by using attribute selection and object selection pane on the right side of lattice drawing pane. After selecting or deselecting name of object or attribute, the new lattice, that corresponds to new selected subcontext is build. In order to include into selection all objects(attributes) use "Select all objects" ("Select all attributes") buttons at the bottom of corresponding panes.

**Warning:** building lattice of subcontexts leads to destruction of information about previous drawing. Please store the image or create a lattice snapshot, if you obtained the useful result after some work.

### Creating a lattice snapshot

If you have spent some time adjusting the layout of the lattice, it may be well worth to create a copy of the drawing before exploring other alternative ways to layout lattice. Also, one may want to be able to consider several drawings of different subcontexts or to be able to consider several alternative drawings of lattice.

In order to create a “snapshot” of the current drawing of lattice (i.e. the exact copy of current drawing of lattice), press “Store current lattice as a view” button on lattice visualization pane toolbar. After that the copy of the drawing will be created and will be shown in the document tree.

### Displaying the lattice statistics

For the currently computed lattices, it is possible to display statistics about its different characteristics:

**Concept count** the number of concepts in the current lattice.

**Edge count** the number of edges in the current lattice.

**Lattice height** the length of the maximal descending path from lattice unit element to the lattice zero element.

**Lattice width estimation** the lower and upper bounds of the lattice width. The lower bound estimation is computed as a maximal count of elements in one layer of the lattice (this value is always less or equals to the lattice width). The estimate of the upper bound is “worse” in sense of precision, and equals to *concept count - lattice height* .

## Working with implication bases

### Calculating Duquenne-Guigues base

In order to find so called Duquenne-Guigues base of implications, that holds in context, one should press button “Calculate Duquenne-Guigues base of implications ” on the main toolbar. The main feature of Duquenne-Guigues base of implications is that this base has a minimal possible number of implications among all possible bases of implications, that holds in context.

Implications, that appears in “Implication sets” pane, has the following format:

*No <Number of objects> Premise ==> Conclusion.*

*No* simply means number of implication in list.

*Number of objects* shows, for how much objects implication holds.

*Premise* and *conclusion* are usually list of attribute names, that occur in premise (conclusion). Also, premise can be “{}”, that means, that this implication has empty premise and holds for all objects from context.

Implication can be displayed in one of two colors: blue or red.

Blue colors means, that there are objects in context, that supports this rule.

Red color means, that there are no objects, that support implication, and usually such implication mean, that set of objects, that contained in premise, doesn’t occur together in context. Also such implication includes all attributes from context among its attributes.

### Searching for associations

Among association rules, in difference from implication, also non-strict rules are allowed, i.e. rules, for which if premise hold, conclusion doesn't necessarily hold - it is true only for some percent of all objects, that are covered by premise of rule. The base of association rules consist of two parts - that base of strict rules (Duquenne-Guigues base) and base of approximate rules (so called Luxenburger base).

ConExp allows to calculate base of association rules. In order to do this, one should press "Calculate association rules" button on main toolbar.

The display format of association rule is a small modification of the format for implications. It is:

*No <Number of objects, for which premise holds> Premise =[Rule confidence]=><Number of objects, for which premise and conclusion holds> Conclusion.*

Also, in addition to red and blue colors, that are used in display of implications, green color is used for approximate, not strict rules.

### Performing attribute exploration

The problem of implications, that are calculated for some context, is that they holds only for objects from context, and don't generally hold for all object from domain of interest. In order to overcome this deficiency, *attribute exploration* procedure should be used.

Attribute exploration is a interactive procedure, in which program asks question about dependencies between different attribute from some fixed set of attributes. If expert confirms, that such dependency generally holds, it should answer "yes", if he rejects dependency, that expert should provide counterexample. If expert answered correctly on all questions, than after the end of attribute exploration procedure he will get the set of all implications, that describes dependencies between different attributes in domain of interest.

Attribute exploration procedure can start from empty context, where only attributes are specified, or from context, where some objects already described.

In order to start attribute exploration procedure, one should press button "Start attribute exploration" on main toolbar.

Then the first question is asked, and user should either confirm it, either reject it, either stop attribute exploration procedure. If user rejects question, than other dialog would appear, that would ask to provide a counterexample.

## Users mailing list

We always glad to hear your feedback about program.

The best place to give feedback and ask questions about ConExp is the ConExp user's mailing list: **conexp-user@lists.sourceforge.net**

## ConExp's team

Currently, team of developers consists of :

- Dr. Serhiy Yevtushenko - initial and chief developer
- Tim Kaiser
- Julian Tane
- Dr. Sergei Objedkov

The documentation team also includes:

- Joachim Hereth-Correia
- Heiko Reppe

If you would like to join development, you are welcome.

## References

[1] Serhiy A. Yevtushenko. *System of data analysis "Concept Explorer"*. (In Russian). Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia, 2000.

[2] B. Ganter and R. Wille. "*Formal Concept Analysis:Mathematical Foundations*", Springer-Verlag, 1999

[3] ComImp - http://www.mathematik.tu-darmstadt.de/~burmeister/

[4] http://www.math.hawaii.edu/~ralph/LatDraw/

[5] http://www.sourceforge.net