developers (and application users as well, of course) such as built-in support for undo/redo and clipboard operations.

All these operations use built-in serialization functionality provided by underlying wxXmlSerializer library [5] in default so no additional programming overhead is required. The operations can be simply invoked by following `wxSFShapeCanvas` member functions:

- `Undo()`, `Redo()`, `Copy()`, `Cut()` and `Paste()`

There are also set of member functions defined in shape canvas class which allow user to determine whether requested operation has a sense at the moment (i.e. any shape is selected or the clipboard contains data understandable for wxSF library). The functions are following:

- `CanUndo()`, `CanRedo()`, `CanCopy()`, `CanCut()` and `CanPaste()`.

Undo/redo operations are internally provided by so called history manager (encapsulated by `wxSFCanvasHistory` class) which can be used for further tuning of their functionality like setting of a history depth (number of stored canvas states) or clearing of whole history. Reference to the history manager can be obtained from shape canvas by `wxSFShapeCanvas::GetHistoryManager()` member function.

4.7 Printing and image export

In addition to the undo/redo and clipboard operations the library has also printing and image exporting capabilities. All these operations are encapsulated by `wxSFShapeCanvas` class (in cooperation with set of other helper classes) and can be simply invoked by its member functions like `Print()`, `PrintPreview()` and `SaveCanvasToBMP()`.

Moreover, printing can be tuned in more details by `SetPrintHAlign()`, `SetPrintVAlign()` and `SetPrintMode()` functions in application source code or interactively by an application user via dialog window which can be displayed by using `PageSetup()` function.

Shape canvas content can be also exported to a bitmap file (in BMP format) by using `SaveCanvasToBMP()` function which will make a snapshot of current canvas drawing.

5 Conclusion

As can be seen from the document and given examples, the wxShapeFramework software library has sufficient potential for effective development of various software applications which use diagrams or other form of visual communications. Note that only very small fraction of all the functions provided by the library has been discussed in this paper. For deeper understanding of its principles and potential we would recommend to go through the library reference documentation and sample projects.

The library can be freely obtained from SourceForge.net software repository [1] and wxCode repository [6] and is distributed under wxWidgets license [4] so it can be used for both open-source and commercial projects without any restrictions. Up to the present day the library has been downloaded more than 4500 times and only few bugs and patches were reported by the users so it can be regarded (despite its relative youth) as sufficiently mature software project. Of course, the development of the wxSF is still in progress so new features and improvements are continuously included to fulfil all requirements of modern cross-platform diagram software library.

6 Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under grant No. MSM 7088352102.

7 References

[1] wxShapeFramework library website, 2008: http://sourceforge.net/projects/wxsf

[2] Smart, J., Hock, K. *Cross-Platform GUI Programming with wxWidgets*, Prentice Hall, 2006

[3] wxOGL code repository at wxCode website, 2008,: http://wxcode.sourceforge.net/showcomp.php?name=ogl

[4] wxWidgets license documents, 2008: http://www.wxwidgets.org/about/newlicen.htm

[5] Bližňák, Michal, Dulík, Tomáš, Vašek, Vladimír, *A Persistent Cross-Platform Class Objects Container for C++ and wxWidgets* , WSEAS TRANSACTIONS on COMPUTERS, Issue 5, Volume 8, May 2009, p.778-787, ISSN 1109-2750.

[6] wxCode repository (official website for wxWidgets add-ons), 2010, https://sourceforge.net/projects/wxcode/files/