Software Architecture Case Study Report for

LibreOffice

Team # 25; Members: Anirudha Ramesh(20171088), Chirag Shilwant(2020201061), John Daniel Mathew(2020201005)

Date: 18/4/2021

1. Introduction

LibreOffice is a community-driven and built software that is a project of The Document Foundation, a non-profit organization. LibreOffice is the most actively developed OpenOffice.org successor project and is free and open-source software based on OpenOffice.org (commonly known as OpenOffice). LibreOffice is created by users like us, who believe in the values of Free Software and the freedom to share their work with the rest of the world. The LibreOffice suite includes programs for word processing, spreadsheet creation and editing, slideshow creation and editing, database function, and mathematical formula composition. It comes in 115 different languages. LibreOffice uses the Open Document Format for Office Applications (ODF), or OpenDocument, as its native file format for all of its applications. ODF is an international standard established jointly by the International Organization for

Standardization (ISO) and the International Electrotechnical Commission (IEC). LibreOffice is available for Microsoft Windows, macOS, Linux, Android, iOS, and Chromebooks, among other platforms. LibreOffice Online, an online office suite that includes the applications Writer, Calc, and Impress, is also available. Most common Linux distributions provide LibreOffice as the default office suite. It is the most actively developed open-source software. Officially intended for personal use, LibreOffice is available in a free and open source "Community" version. This is the complete suite, not a shortened edition. TDF's corporate partners will also provide LibreOffice versions of enterprise support.

History

Back then, it was known as StarOffice 3.1, not LibreOffice. "StarWriter, the ancestor of the LibreOffice suite, was created as proprietary software by Marco Borries, a German student, to write his high school final thesis," according to the Document Foundation. To build the software, he started a company called Star Division. Sun Microsystems purchased Star Division for $73.5 million in 1999, renamed the software to OpenOffice.org, and made the code open source. For personal use, anyone may download the office suite for free. "From version 1.0 to version 3.2, the software was built under Sun's stewardship for almost ten years," the Document Foundation states. It began with a dual license—the LGPL and the proprietary SISSL (Sun Industry Standard Software License), but as of version 2.0, it was just the LGPL. When Oracle bought Sun Microsystems in 2009, there was a lot of speculation about what would happen to OpenOffice. In 2010, the Document Foundation forked the project, renaming it LibreOffice. Oracle stopped developing OpenOffice in 2011 and handed the project over to the Apache Foundation. Over 95% of the previous OpenOffice group rallied behind the newly autonomous and community-led Document Foundation. LibreOffice continues to grow, develop, and prosper as a result of their contributions.[1]

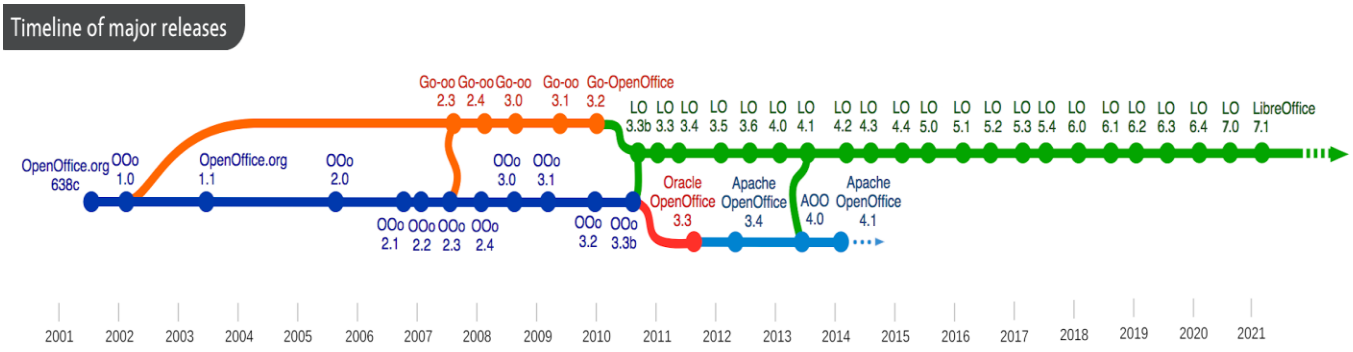


Fig. LibreOffice Timeline

The entire history of LibreOffice can be understood by the table shown below.

Year	Releases
1985	Marco Borries's Star-Writer I is released, originally for CP/M on the Amstrad CPC and later ported to DOS. StarDivision is established in Lüneburg, Germany's northernmost city.
1991	StarWriter 5.5 was released, including StarWriter Compact 2.0
1992	StarOffice 1.0, which combines StarWriter, StarBase, and StarDraw, was released.
1999	Sun Microsystems pays $73.5 million for StarDivision and buys it.
2000	StarOffice 5.2 was available for personal use as a free download. Sun Microsystems released OpenOffice.org as an open-source version of StarOffice.
2001	OpenOffice.org XML, the default file format used by the suite, is contributed to the OASIS consortium - this is later developed into the Open Document Format (ODF)
2002	OpenOffice.org 1.0 was released.
2009	Oracle pays $7.4 billion for Sun Microsystems and buys it.
2010	Members of the OpenOffice.org community announced the Document Foundation. It was a non-profit, self-governing, democratic foundation dedicated to the advancement of open-source office software (incorporated in Berlin in 2012). LibreOffice, a fork of OpenOffice.org, was the foundation's main project.
2011	LibreOffice 3.3 was now available. Oracle abandons OpenOffice.org production and contributes the trademarks to the Apache Software Foundation. In Paris, France, the first LibreOffice Conference was held.
2013	LibreOffice 4.0 and LibreOffice 4.1 was released.
2014 to 2020	Different versions of LibreOffice were released with new addons.
2021	Script Forge macro tools are now available in LibreOffice 7.1, which includes physics-based animations in Impress, a Styles Inspector in Writer, and physics-based animations in Impress.

2. Requirements and Qualities

Since the roots of today's LibreOffice started setting in decades ago, even prior to the announcement of MS Office, the requirements and the expectations of the software has evolved massively.

However, fundamentally, some of the basic functional requirements have remained same throughout the years. Today, LibreOffice aims to provide:

1) Word Processor (Writer)

2) Spreadsheet with various possible mathematical operations (Calc)

3) Presentations (Impress)

4) Vector graphics (Draw)

5) Database Management (Base)

6) Formula Editor (Math)

As indicated by the names in the bracket alongside, each of these core functionalities are primarily assigned to different applications within the platform, though each platform is not restricted to only these functionalities.

Looking into these applications in more detail,

1) Writer:

Key Functional Requirements catered to are:

a) Creation of text documents whilst using multiple language scripts.

b) Insertion of tables, shapes/objects, graphics

c) Export file into various formats like HTML, PDF, MS Word, XML etc.

d) Saving files for future updates. (Common to all below)

e) Additionally, also connects to email client allowing sending documents via email directly as well provide access to various mathematic functions.

2) Calc:

Akin to MS Excel, calc is a workhorse that aids countless industries.

Key Functional Requirements catered to are:

A) Allowing charting of tables, accounts etc.

B) Providing complete suite of advanced analysis tools to assist monitoring and decision making. It includes over 300 functions for financial, statistical, and mathematical operations, among others.

C) Providing analysis models for various scenarios via the scenario manager allowing non-linear analysis alone.

D) Generating charts (2D and 3D) which can be integrated into other applications in the Libre Office suite.

E) Exporting Spreadsheets in various formats like CSV, PDF etc.

3) Impress:

Key Functional Requirements catered to are:

A) Providing a system that allows design and display of slides.

B) Multimedia presentation tools such as special effects, animation, various fonts and text styles, and drawing tools.

C) Allowing embedding of sound and video clips into slides.

D) Integration with the advanced graphics capabilities of LibreOffice Draw and Math components.

E) Allowing export into different formats including being compatible with MS PowerPoint format.

4) Draw:

Key Functional Requirements catered to are:

A) Allowing the creation of vector graphic art by using both provided building blocks or drawing lines/curves by marking points.

B) Drawings created must be made integrable into all other LibreOffice applications.

C) Export into various formats like png, jpeg, pdf, flash etc.

D) Import and work with various formats like png, jpeg, pdf, flash etc.

5) Base:

Key Functional Requirements catered to are:

A) Creation of Database, and Tables via UI.

B) Support for responding to queries, view, edit, insert etc. via UI.

C) Analyze and Edit Relations using a diagram view.

D) Import and Export of databases to and from a variety of formats.

6) Math:

Key Functional Requirements catered to are:

A) Creation and editing of complex equations using various mathematic and scientific symbols and notations.

B) Produce outputs that can be used/integrated in other LibreOffice applications.

C) Export into MathML and other formats for cross platform support outside LibreOffice as well.

Quality Attributes:

In addition to these functions requirement LibreOffice also aimed(aims) in creating a **cross-platform** suite that allows these utilities to be made more indifferent to the systems they are on.

Extensive language support also allows usage of this suite across all regions of the world.

The design also wants to feciliate easy **accessibility** to users by keeping a clear, **consistent interface** across applications. LibreOffice online takes this to next level as it allows storage and editing of documents on the move, but this brings to the fore the issue of successful *scalability, availability, security and reliability* to support a large number of users remotely.

As already indicated in the functional requirements, there has been an emphasis on **integration and inter-usability** amongst different applications in the LibreOffice suite meant to provide a complete user experience within the suite itself.

At the very heart of this project are the values forged from Open Source software practices, and as an extension of this, LibreOffice uses OpenDocument, an XML (eXtensible Markup Language) file format developed as an industry standard by OASIS (Organization for the Advancement of Structured Information Standards). These files can easily be unzipped and read by any text editor, and their framework is open and published. This **prevents vendor lock-in**.

In addition to its native OpenDocument formats, LibreOffice includes support for opening and saving files in many common formats including Microsoft Office, HTML, XML, WordPerfect, Lotus 1-2-3, and PDF thus providing a high degree of **file compatibility**.

Apart from this, there is without a doubt a great emphasis placed on **reliability, performance, capacity etc**. The ability to create, maintain and manage large files (I.e., capacity) is one of the key reasons for the very creation of this software.

Though it is extremely hard to locate an 'ordering' of importance amongst these key attributes, one might assume with a fair certainty that the open-source principles of providing great applications at no cost for people across the board lie at its core.

In the above section, we can also see how a lot of these attributes are visible in the delivered products. Elucidating some of them below:-

1) Capacity: Large File sizes possible

2) Cross-Platform – Works across various hardware (even from different eras), multiple operating systems like Mac, Linux, Windows.

3) Accessibility - LibreOffice online and usage of consistent interfaces across applications in the suite.

4) Reliability, Performance – The offline products have been used for decades by millions without too many issues, and this alone speaks of how well these qualities have been brought in.

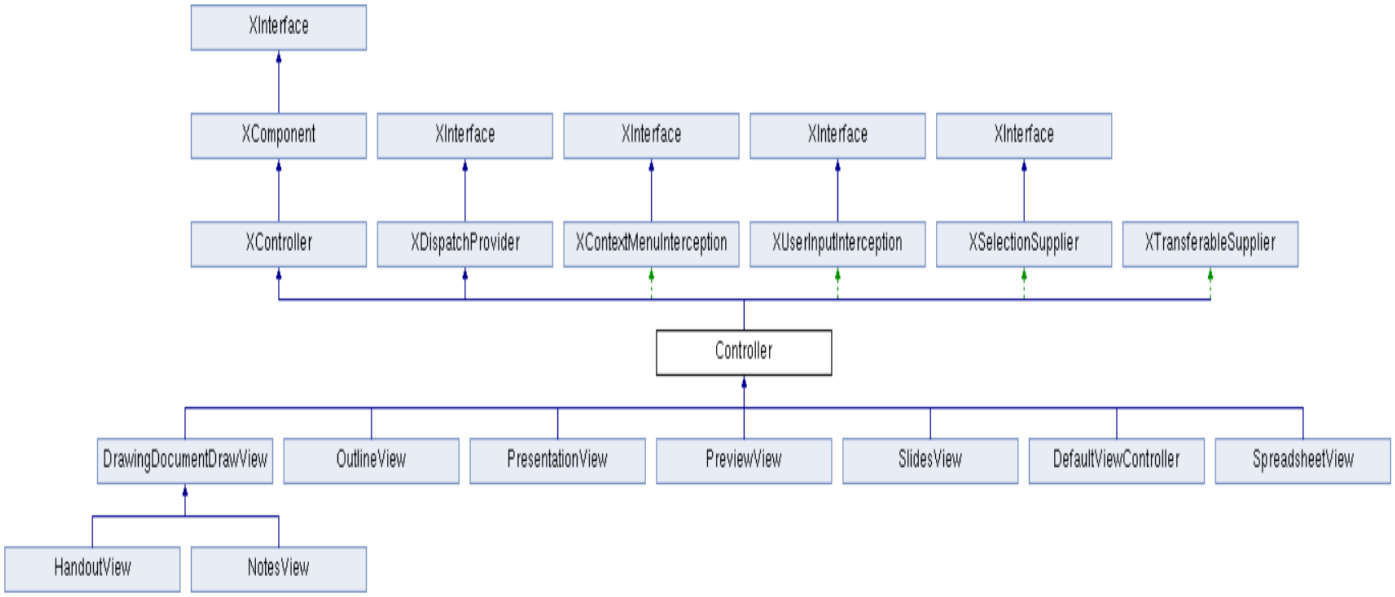
Various other such examples have been listed in the above section. Overall, we can see that there has been a clear commitment towards these desired quality attributes, and these have largely been successfully delivered as well.

3. Architectural Solution

The Architecture of LibreOffice comprises of Frame-Controller-Model. Model-View-Controller (MVC) is a well-known application paradigm that divides three application areas: document data (model), presentation (view), and interaction (controller) (controller). The Frame-Controller-Model (FCM) paradigm in Apache OpenOffice is a related abstraction. The Frame-Controller-Model model is similar to Model-View-Controller in several ways, but it serves different purposes; as a result, it's better to view Frame-Controller-Model separately from Model-View-Controller. In Model-View-Controller and Frame-Controller-Model, the model and controller are very different.

In Apache OpenOffice, the Frame-Controller-Model paradigm divides three implementation areas: the document object (model), the screen interaction with the model (controller), and the controller-window linkage (frame).

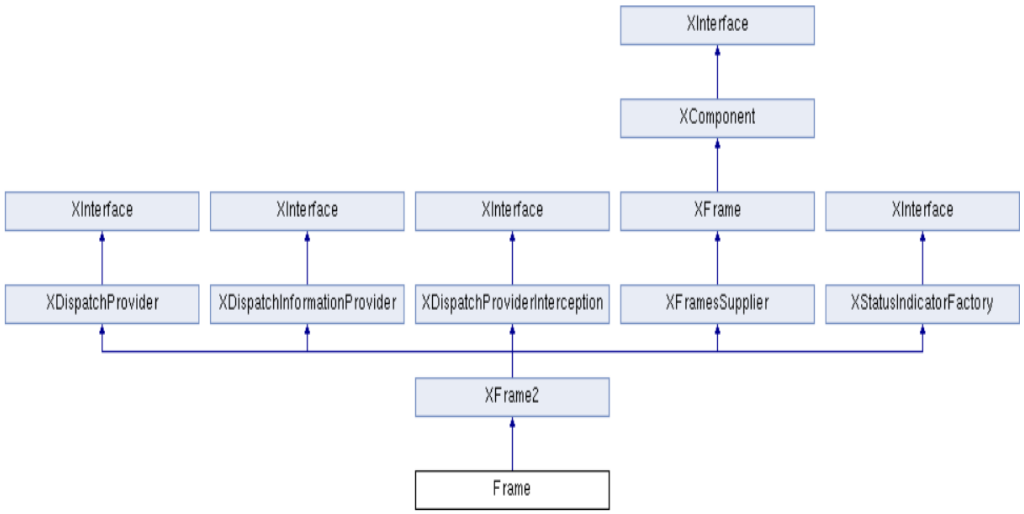
Inheritance diagram for Controller:



Frame

Frame is the bridge between controller and the window. It is also through which bidirectional communication between UI and controller happens

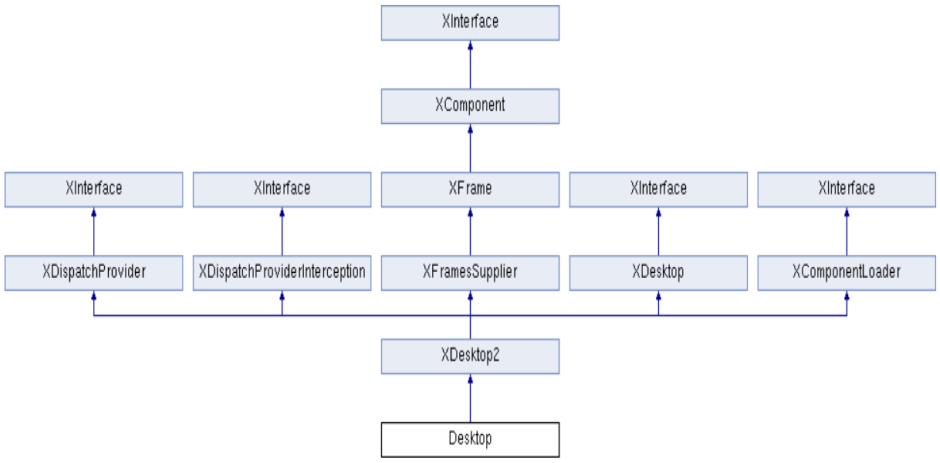
Inheritance diagram for Frame:



Desktop

Desktop is the root frame

Inheritance diagram for Desktop:



Components

Components

A frame attaches components to two windows

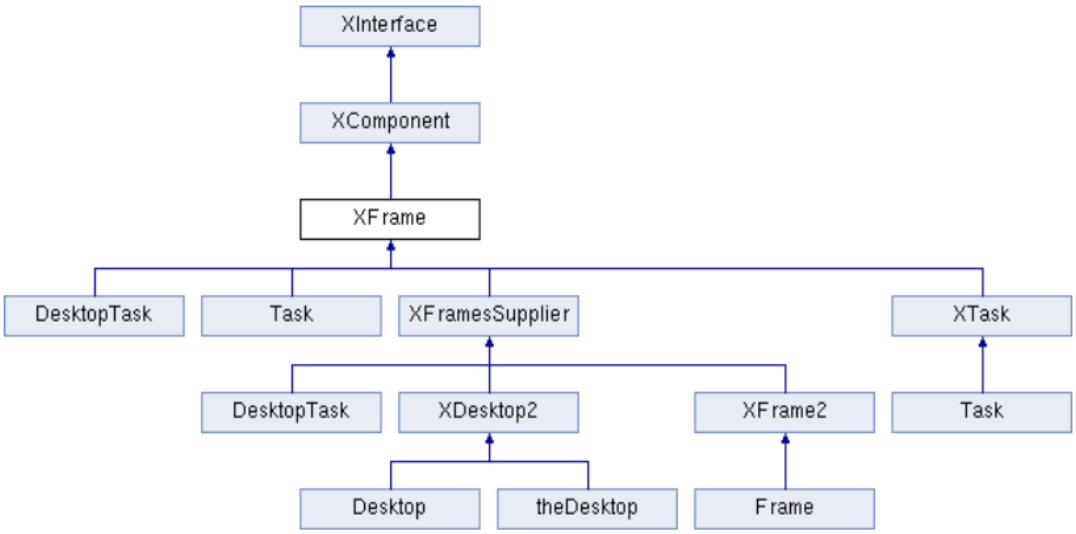
1.) component window

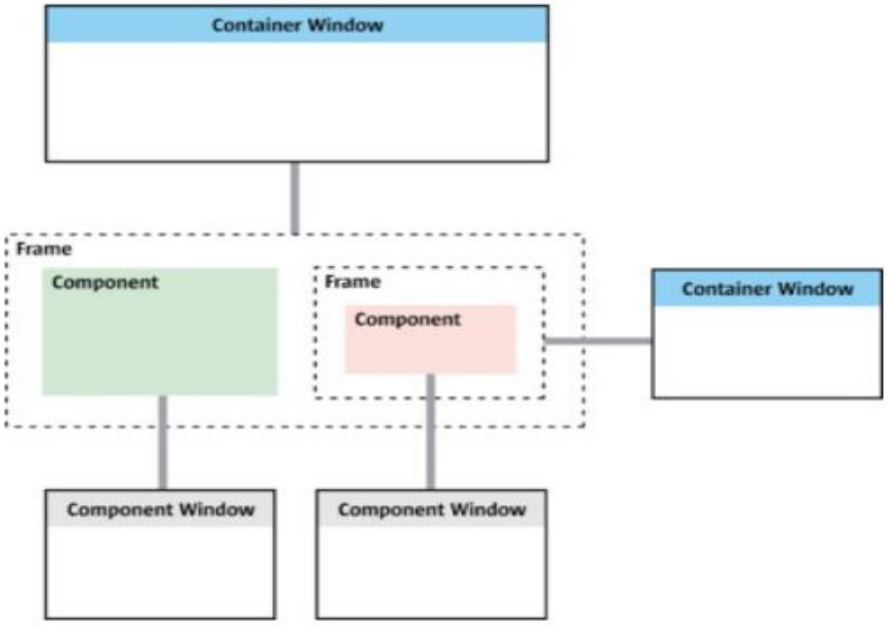
2.) container window

A frame can contain several components .

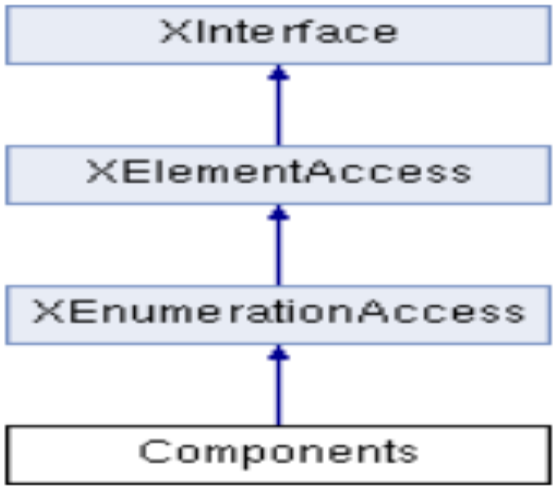
Frame Hierarchy

A frame can have multiple components. It can have another frame within itself.





Component Hierarchy



The aim of Frame-Controller-Model is to have three interchangeable parts that can be used with a window system that can be swapped out. Without modifying the model or the frame, a new controller can be written to present an existing model in a new way. Since a controller is dependent on the model it introduces, it is possible to write a new controller for a different model. [2] Developers don't have to worry about the frame or the underlying window management framework as they implement new models for new document forms. However, since there is no default controller, a suitable controller must be written as well. It is possible to use a single frame implementation for any possible window in the entire Apache OpenOffice application by keeping all window-related features separate from the frame. Let's see how this architecture was achieved by looking at its codebase.

Code Structure

Let's understand the Module in three levels: lowest level, middle level and upper level.[3]

1. **Lowest Level**
   - Sal (at the bottom)
     - The system abstraction layer.
     - 'tools' is an obsolete internal ~duplication of this module
   - Salhelper
     - wrapper code around
   - Registry
     - used to keep interface descriptions
   - Store
     - obsolete & irrelevant.

- Unoidl
  - used to create / compile interface descriptions: an IDL compiler.
- Cppu
  - Basic UNO forms and infrastructure for C++ are implemented.
- Xmlreader
  - a very basic XML parser.
- Cppuhelper
  - luggage to bootstrap UNO, create UNO components etc.
- Ucbhelper
  - Universal Content Broker (ucb) C++ wrapper / helper classes.
- Comphelper
  - lots of good C++ stuff for using UNO – not stable enough to go into the URE.
- Jvmfwk
  - Java / UNO integration

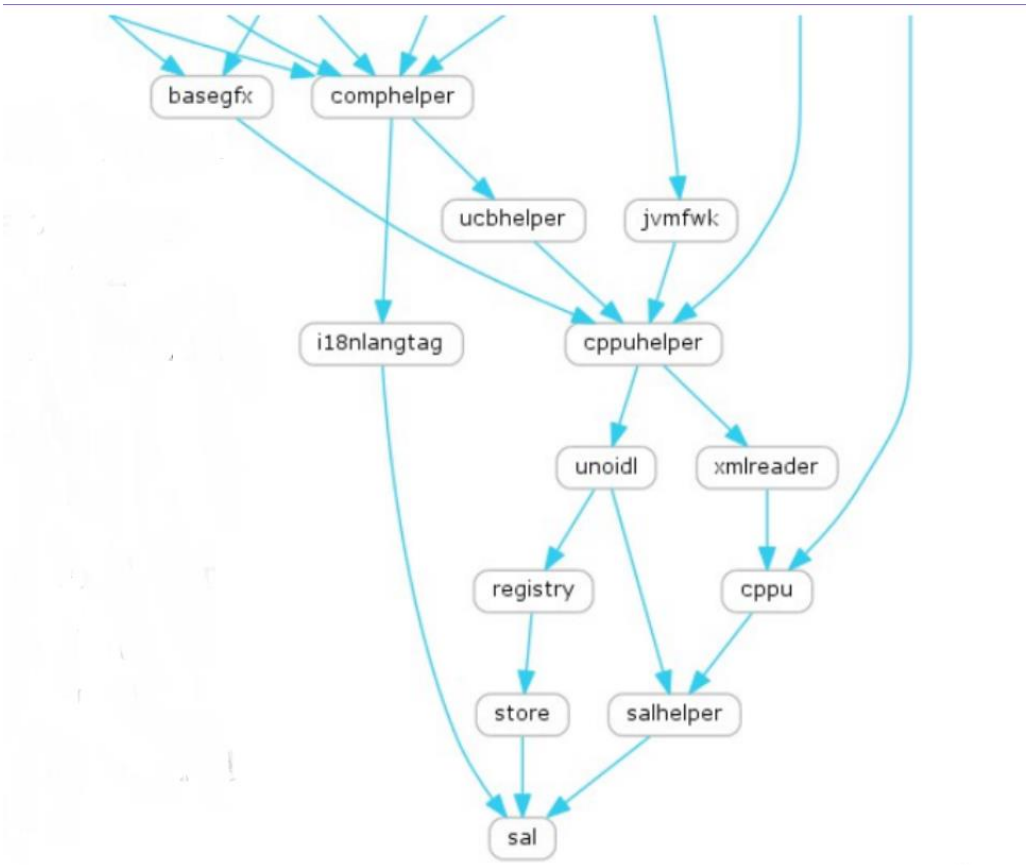


Fig. Module Overview - Lowest Level

2. **Middle Level**
   - Basegfx
     - algorithms / graphic types etc. for basic graphics.
   - Tools
     - SvStream – internal stream type – vs. UCB vs. sal/ file pi
     - Color COL_RED etc.
     - INetURLObject – canonical URL handling
     - A total grab-bag of things
     - Date / Time classes
     - SolarMutex (the big lock)
   - Cppunit
     - all our tests are ultimately cppunit tests though this is an external module.
   - Unotest
     - low level testing of simpler / UNO infrastructural pieces. Bootstrap UNO enough to be able to test filters, components etc.
   - Test
     - helpers for testing standard interfaces, more advanced tests brings UCB bootstrap (for streams), VCL initialization, graphic filter pieces etc.
   - Svl
     - non-graphical (no VCL dependency) pieces originally from svtools/ or sfx/ eg.
   - Sot
     - handles OLE2 / compound file storage for binary documents.
   - Unotools
     - C++ helpers for using UCB.
   - Toolkit
     - a particularly thin & horrible UNO API wrapper with Model/View flavour on top of vcl.
   - Canvas
     - alpha transparent, antialiased UNO rendering API – more modern rendering than VCL, primarily used by slideshow.
   - Cppcanvas
     - C++ wrappers to make using the canvas less bad.
   - Xmlscript
     - XML serialisation of (orrible) basic dialogs which wrap the toolkit pieces for in-document scripting / macro dialogs.
   - Connectivity
     - UNO implemented database drivers for all manner of backends.
   - Sax
     - wrapper of libxml2 – providing an UNO sax API for parsing XML files, and an XFastParser for tokenising them.

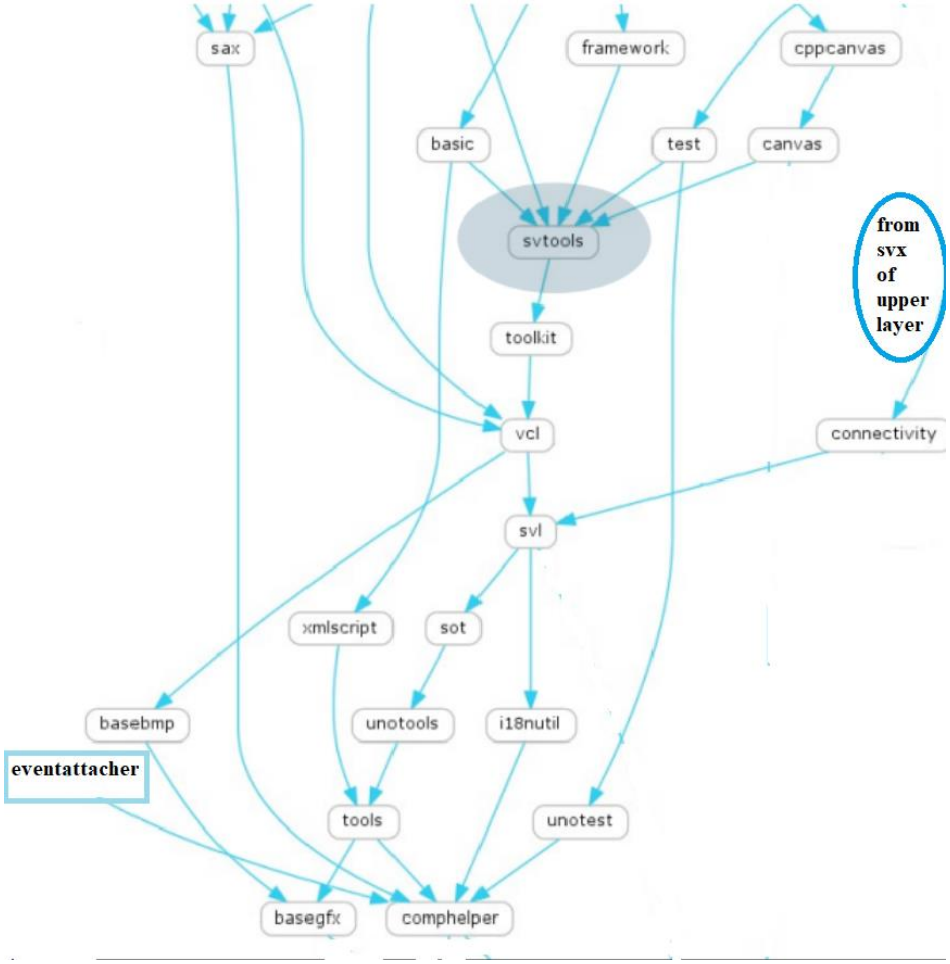


Fig. Module Overview - Middle Level

3. Upper Level

- Framework
  - manages docking, toolbars, menus, status bar, sidebars, task-panes.
- Sfx2
  - works closely with framework, core of the app
  - Document meta-data dialogs
  - Template management
  - Shared style pieces.
- Formula
  - nominally shared code extracted from calc (sc) for use in reportdesign.
- Avmedia
  - Audio / Video media – multimedia abstraction over DirectX, quicktime, gstreamer.
- Package
  - ZIP file compress / decompress, also handles manifest files in the .zip with UNO stream / storage interfaces.

- Desktop
  - legacy name, StarOffice 5 had a 'desktop' complete with 'Start' menu etc.
- Sd
  - Star Draw - For Drawings + Presentations
- Sw
  - Star Writer - For Word processor
- Sc
  - Star Calc - Spreadsheet
- Xmlsecurity
  - XML document encryption and signing used for ODF.
- Vbahelper
  - helper code for implementing VBA / macro interoperability with MS Office.
- Xmloff
  - ODF file filters and helpers to load / save our model to/from ODF.
- Filter
  - meta-data to manage, register and auto-detect filters.

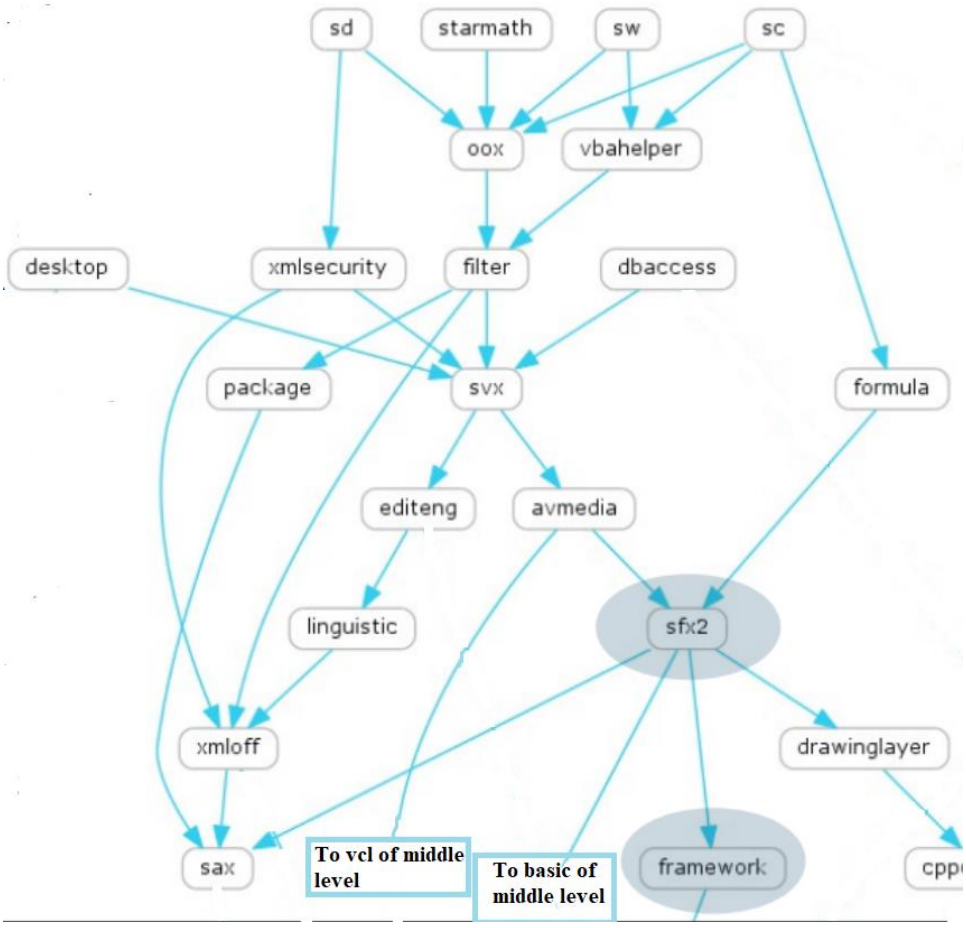


Fig. Module Overview - Upper Level

4. Summary

LibreOffice is the most actively developed OpenOffice.org successor project and is free and open source software based on OpenOffice.org (known as OpenOffice). LibreOffice is created by users like us, who believe in the values of Free Software and the freedom to share their work with the rest of the world. LibreOffice has shown the benefits of open-source development over proprietary development in a variety of ways, most notably by standardizing on the Open Document Format (ODF). According to the Document Foundation, ODF is the only true standard format for office documents available today.[4] Thanks to the Document Liberation Project's volunteer developers, LibreOffice will open a broad range of document forms. Now, Microsoft Office is not required to open.doc,.docx,.ppt,.pptx,.xls, or.xlsx files. It can import Keynote, Pages, and Numbers from Apple. It can also read files created with Microsoft Works, WordPerfect, and Lotus 1-2-3.

This kind of interoperability isn't just about liberty; it also has financial advantages. LibreOffice is available in over 110 languages, allowing it to enter "the greatest number of people in their native language." The Document Foundation states that "in some cases, this language represents a language minority, such as Guarani in South America, which is spoken by many in Paraguay and regions of Argentina, Bolivia, and Brazil." According to the Document Foundation, there are 200 million active LibreOffice users worldwide, with around 25% of them being students and 10% being Linux users who usually find LibreOffice part of their preferred distribution.

Making the code open source encouraged the growth of a self-governing, global community founded on meritocratic principles. People from all over the world are included in the group, which covers six continents. The Document Foundation reports that 1,000 contributors are participating on a weekly basis, with another 4,000 people available as required.

Strengths

- Full-featured - It includes almost all you'd find in a professional office suite, including a word processor, spreadsheet, slideshow, painting, and database.
- Follows industry guidelines - for example, the spreadsheet uses the same formulas as Excel.
- Flowcharts with arrows that automatically attach to boxes, following smooth lines, are among the features available in the Draw software that you won't find elsewhere (at least not easily).
- Actively maintained, with regular and valuable updates.
- It is compatible with all three big PC platforms: Windows, Linux, and Mac (including official application stores for these systems).

Weaknesses

- The menus are outdated; although it is fully functional, certain resources are hidden inside dialogues that must be found under sub-sub-menus, and so on.
- If it becomes noticeable to clients or other stakeholders, it may make a bad impression; this isn't a fault with the suite, but first impressions are important.
- Despite the large number of templates available, it's obvious that this platform is primarily supported by developers and Linux users, who, in contrast to MacOS or Windows, do not have as many graphic designers.

5. References

[1] https://www.libreoffice.org/

[2] https://api.libreoffice.org/docs/idl/ref/servicecom_1_1sun_1_1star_1_1frame_1_1Frame.html

[3] https://wiki.openoffice.org/wiki/Documentation

[4] https://www.documentfoundation.org/

[5] https://www.libreoffice.org/about-us/libreoffice-timeline/

[6] https://documentation.libreoffice.org/assets/Uploads/Documentation/en/GS5.2/HTML/GS5201-IntroducingLibreOffice.html#__RefHeading__22007_697337876

[7] https://www.guru99.com/functional-vs-non-functional-requirements.html

6. Effort

The amount of time (in hours) spent on this deliverable by each team member:

Anirudha Ramesh: 15 hours

Chirag Shilwant: 15 hours

John Daniel Mathew: 15 hours