

OpenOffice.orgに対する提案： 統合開発環境による開発と拡張可能な簡易エディタ

メタデータ	言語: Japanese 出版者: 大阪市立大学創造都市研究科情報学専攻 公開日: 2024-09-09 キーワード (Ja): キーワード (En): 作成者: 矢吹, 幸治 メールアドレス: 所属: 大阪市立大学
URL	https://ocu-omu.repo.nii.ac.jp/records/2016126

大阪市立大学大学院
創造都市研究科 電子紀要

OpenOffice.org に対する提案
— 統合開発環境による開発と
拡張可能な簡易エディタ —

Proposals for OpenOffice.org
— Development of Integrated Development Environment and
Extendable simple Editor —

2005年4月
大阪市立大学大学院 創造都市研究科
都市情報学専攻 情報基盤研究分野
M03UC530 矢吹 幸治 (YABUKI, Yukiharu)

概要

本論文は、コンピュータソフトウェアのソースコードやバイナリーコードを世界規模で共有し、開発するというスタイルを取るオープンソースソフトウェアの一つである Office Suite の OpenOffice.org をターゲットとしている。

現在の OpenOffice.org は、コマンドを入力してコンピュータに指示する Character User Interface(以下, CUI), で開発がなされている。本論文では、統合開発環境 (Integrate Development Environment, 以下 IDE) として、オープンソースソフトウェアの IDE である Eclipse を使用する。Eclipse を使用して OpenOffice.org の実行形式の生成方法について論述する。

次に、OpenOffice.org の ワードプロセッサ Writer に含まれる、最小限の機能セットを検討する。検討した最小限の機能セットを ooEditor として定義して、後から必要な機能を付け加えるモジュールとしてプラグイン機能について論述する。

ooEditor に機能を組込む際に、各プラグインの持っている機能を調整する仕組みとしてプラグイン間に依存関係を記述する。

目次

1	はじめに	3
2	Open Source とは	3
2.1	Open Source 概論	3
2.2	Open Source の定義	4
3	OpenOffice.org の概要	4
3.1	OpenOffice.org とは	4
3.2	OpenOffice.org における開発	5
4	開発手法に対する問題	6
4.1	OpenOffice.org の問題	6
5	IDE と Eclipse	7
5.1	IDE の長所と短所	7
5.2	Eclipse とは	7
6	Eclipse による OpenOffice.org の開発	7
6.1	Eclipse を使う利点	7
6.2	Eclipse 開発環境	8
6.3	Eclipse による OpenOffice.org のビルド方法	8
6.4	問題点	8
7	ooEditor の開発	8
7.1	ooEditor 開発の意義	9
7.2	ooEditor とは	9
7.3	最小の機能とは	9
7.4	ooEditor 仕組の概観	10
7.5	ooEditor のプラグイン機能	10
7.6	ooEditor プロトタイプ作成	11
8	結び	12
	謝辞	12
	参考文献	12

1 はじめに

インターネットの登場により大きく発展したソフトウェアにオープンソースソフトウェアがある。オープンソースソフトウェアは、インターネットの上に開花したコンピュータ文化である。ソフトウェアのソースコードおよびバイナリーの公開により、インターネットの上に知のプールができており、個人、地域、国家などのあらゆるレベルでの利用が進んでいる。オープンソースソフトウェアでは、多くの人が、それぞれの能力に応じて参加してソフトウェアを作成する所に意義がある。文献 [1] では、バザールと呼んでいる形式が示されている。

オープンソースソフトウェアは、欲しい人が欲しい機能を作り込んで行くのが基本である。今後オープンソースソフトウェアが活用されるにしたがってオープンソースソフトウェアに求められるものも量が増え、質が高くなる。量と質に求められるものが多くなると解決する人間も多様な技能を持った開発者が多数必要となる。

例を挙げると、オープンソースソフトウェアが Enterprise 系でのニーズに応えるには、企業の要求や企業の文化の知識・経験がある開発者が必要である。また、別の例では海外で開発されているオープンソースソフトウェアを国際化の枠組を使い日本語で利用可能とするためには、ソフトウェアの国際化、多言語化に関する知識・経験がある開発者が必要である。

オープンソースソフトウェアは、特定のソフトウェア分野を対象とするものではない。ソフトウェアの分野として確立している分野でオープンソースソフトウェアが求められているのに開発者が少ないのであれば、自らその分野に向う開発者が現われる事が期待される。

本論文では、デスクトップ用途の利用で圧倒的なシェアを誇っている Microsoft Office との互換性を持ったオープンソースソフトウェアの Office Suite として、OpenOffice.org をターゲットとして議論を行う。OpenOffice.org の開発手法に新たな手法を提案することで、OpenOffice.org の開発に多様な開発者の参加を促すことが可能になると考えている。

具体的にはオープンソースソフトウェアである OpenOffice.org の開発手法および OpenOffice.org の基本的な機能をプラグイン形式で拡張するという ooEditor の提案を行う。

まず、第 2 章で、オープンソースについて述べる。次に第 3 章は、オープンソースの Office Suite である OpenOffice.org について概論を述べる。第 4 章は、OpenOffice.org の現在の開発手法について述べる。第 5 章は、オープンソースソフトウェアの IDE である Eclipse について述べる。第 6 章では、OpenOffice.org の開発を行うにあたり、既存のコマンドラインの開発環境と、オープンソースの統合開発環境 (以下 IDE^{*1}) として、脚光を浴びている。Eclipse を比較する。その後、OpenOffice.org の開発に Eclipse を利用した開発手法を適用する提案を行う。具体的には Eclipse から、OpenOffice.org をビルドする方法について述べる。第 7 章では、軽量の ooEditor 作成の意義を述べ、OpenOffice.org の

Writer の部分について最小限の機能セットを検討する。次に ooEditor として機能の選択を行なう。その後、個々人の都合によって必要な機能を追加する手段としてプラグイン機能について提案する。最後に ooEditor のプロトタイプについて述べる。第 8 章は、本論文のまとめになる。

2 Open Source とは

2.1 Open Source 概論

オープンソースソフトウェアとは、単に無料のソフトウェアであったり、プログラムの設計図であるソースコードを閲覧できるだけのソフトウェアではない。

オープンソースは、2 つの方向がある。2.2 節の定義に沿ったソフトウェアを開発し、利用する実利的な領域を指向する方向と、R. M. Stallman 氏が提唱している、「自由なソフトウェア^{*2}」運動という思想を背景に持つ方向である。

ソフトウェアが商業化されるに従い、プログラムの知のプールであるソースコードを外部の人間と共有することができなくなった。ソースコードがあれば、コンピュータの特性上 100% 同じプログラムを再現できる。しかし、コンピュータのプログラムを販売する — 会社によっては、プログラムの販売ではなく、プログラムを実行する権利を利用者へ許諾する。つまりプログラムを利用する権利だけ販売し、自社が開発したプログラムは企業秘密であるという立場である。 — 会社にとっては、自社の商品の価値を下げるという考えがある。ソフトウェアを販売する会社は、ソフトウェア利用者に対してソフトウェアを利用者毎に購入してもらうためにソフトウェアを他人と共有と利用できないように様々な工夫がなされている。

しかしながら、ソースコードやバイナリープログラムはデジタル情報である。これらのデジタル情報は複製のコストが低い。またインターネットによりデジタル情報の流通が容易になり、ソフトウェア作成を共に行なうインフラは整った。そのインフラの上で、多数によるソフトウェアの開発サイクルを高めたものが、文献 [1] にもあるオープンソースの本質である。

もともとは、フリーソフトウェアの共同作業を促進するために、R. M. Stallman 氏が copyleft [5, pp. 143–146] の概念を考えた。copyleft は、プログラムを著作権法の元に置いて、プログラムのコードを利用、変更、再頒布する自由を確保する。というものである。この copyleft の概念は、オープンソースに多大な影響を与えている。R. M. Stallman 氏がこの copyleft の概念をひとつの形にしたのが GNU GPL [13] である。

オープンソースという言葉は、1998 年 2 月 3 日に Eric Leymond や VA Linux 社 (現 VA SOFTWARE [9]) の Larry M. Augustin らがフリーソフトウェアという言葉が「無料」という意味の Free という単語を含んでいるためにビジネスにとって不利に働いている問題を認識していた。オープンソースという言葉は、この問題を解決するために、多くの人に受け入れやすい形のメッセージとして作られた造語 [6, pp. 202] である。オープンソースという言葉ができた経緯については、関係者自身

*1 Integrate Development Environment

*2 Free Software

がインタビューに応じている DVD[8] の中で語っている。

オープンソースソフトウェアのプロジェクトに必要なことは、文献 [7, p187] にもあるように、プロジェクトが存続し、またプロジェクトを発展させるための献身的なユーザコミュニティを持つことである。

2.2 Open Source の定義

オープンソースの定義については、英語による原文 [11] の定義がある。八田真行氏により日本語の参考訳 [12] を参考にして下記に引用する。

(1) 自由な再頒布 (が可能であること), (2) (プログラマが変更可能な) ソースコード (の入手方法の担保), (3) 派生ソフトウェア (の存在の容認, copyleft), (4) 作者のソースコードの完全性 (integrity), (5) 個人やグループに対する差別の禁止, (6) 利用する分野 (fields of endeavor) に対する差別の禁止, (7) ライセンスの分配 (distribution)^{*3}, (8) 特定製品でのみ有効なライセンスの禁止, (9) 他のソフトウェアを制限するライセンスの禁止, (10) ライセンスは技術中立的でなければならない

上記の定義は、Bruce Perens 氏が「Debian フリーソフトウェアガイドライン」という文書を Debian コミュニティの人達と一緒に作りあげた後に、Debian 固有の話題を文書から削除して、この「オープンソースの定義」を作った。

3 OpenOffice.org の概要

3.1 OpenOffice.org とは

OpenOffice.org とは、オープンソースソフトウェアとして提供される Office Suite である。OpenOffice.org は、現在の Office Suite で支配的な地位を占めている Microsoft 社の Office Suite (以下、MS Office) とのファイル互換性を持っている。

オープンソースソフトウェアであることから、ソースコードを閲覧、改変^{*4}、配布できる。また、バイナリも配布できる。

OpenOffice.org は、入手するコストが低く、ソフトウェア内部も確認可能であり、MS Office と互換性もあり、ユーザーへの高い柔軟性を持っている。列挙している、これらの優位点から現在のデファクトスタンダードである MS Office を置き換える本命可能性を持つソフトウェアであり、オープンソースソフトウェアで構成された PC デスクトップ利用を推進する上で大きな力になると期待されているプロジェクトである。

OpenOffice.org ユーザ会のユーザーズガイド [24] によると、OpenOffice.org の Office Suite としての特徴は、下記の 3 つである。

1. 無料で使える Office Suite である
2. MS Office との互換性を持っている
3. さまざまな環境に対応している

^{*3} プログラムに付与された権利がそのプログラムが再頒布された者全てに等しく付与されることの担保

^{*4} 開発元に改変を提案をして受諾してもらうには、Sun Microsystems 社との間に JCA[27] という契約が必要になる。

1については、語弊があるがオープンソースソフトウェアの一側面の部分に光をあてている。

以下では、3.1.1 節にて、OpenOffice.org の動作プラットフォームについて述べる。次に 3.1.2 項にて、オープンソースソフトウェアの OpenOffice.org と Sun Microsystems のソフトウェア製品である Star Suite の関係について明らかにする。最後に 3.1.3 節にて、OpenOffice.org の Office Suite としての機能を述べる。

3.1.1 OpenOffice.org の動作プラットフォーム

OpenOffice.org の動作プラットフォームは、OpenOffice.org オフィシャルユーザーズガイド [24] および、Web Page[26] によると、

- Microsoft Windows
- Linux
- FreeBSD
- Mac OS X^{*5}
- Solaris
- Linux PPC

で動いている、と記述がある。他には、IRIX や VMS など動作するように開発中である。

3.1.2 OpenOffice.org と Star Suite

OpenOffice.org は、Star Office から派生したオープンソースソフトウェアである。文献 [24] の 2 章に記述があるが、初期の Star Office を開発したのは、ドイツの Star Division 社である。Sun Microsystems 社は、1999 年 8 月 Star Division 社を買収して Star Office に関する権利を取得した。日本およびアジア地域では、Star Office という名前は、別の会社が商標を持っているために、名称を Star Suite に変更した。

Sun Microsystems 社は、この Star Suite を自社製品版としてリリースして販売している。OpenOffice.org コミュニティがオープンソース版をリリースしている。両者は、おたがいのソースコードのうちオープンソースになっている部分を共有して開発している。

3.1.3 OpenOffice.org の機能

OpenOffice.org は大別すると以下の 6 つのソフトウェアから構成される。

- OpenOffice.org Writer (ooWriter)
Microsoft の Word とファイルの互換性を持つ、ワードプロセッサである。(図 1)
- OpenOffice.org Calc (oocalc)
Microsoft の Excel とファイルの互換性を持つ、表計算ソフトである。
- OpenOffice.org impress (oointpress)
Microsoft の PowerPoint とファイルの互換性を持つ、プレゼンテーションツールである。
- OpenOffice.org Draw (oodraw)
ドロー用ソフトウェアである。
- OpenOffice.org Math (oomath)
数式エディタである。
- OpenOffice.org HTML Editor (ooweb)
HTML エディタ、オーサリングツールである。

^{*5} 2004/08 ではベータレベル

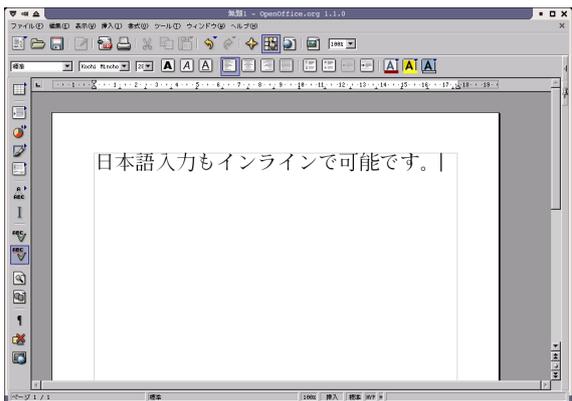


図 1 OpenOffice.org 1.1 Writer

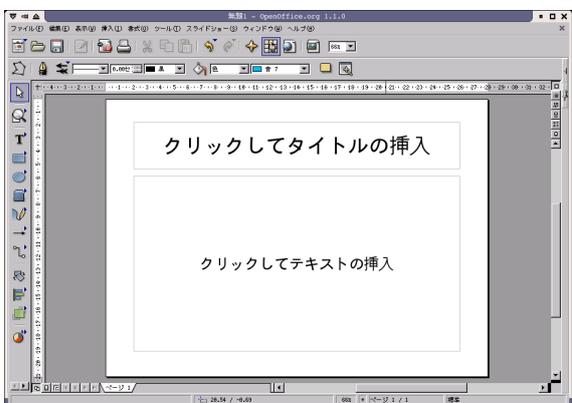


図 2 OpenOffice 1.1 Impress

各ソフトウェアはテンプレートの機能を持ち、ooWrite/ooCalc/ooImpress/ooWeb は、Adobe 社の PDF 形式で出力できる。また、ooImpress および ooDraw では Macromedia 社の Flash Version 4 レベル形式で swf ファイルを出力できる。

3.2 OpenOffice.org における開発

3.2.1 開発体制

OpenOffice.org の開発方法として、オープンソース的な手法という特徴がある。

現在のところ、OpenOffice.org の開発は、Sun Microsystems 社がイニシアチブをとっている。Sun Microsystems 社で、OpenOffice.org の開発の中心となっているのは Star Division 社の所在地であったドイツのハンブルグである。

文献 [24] の開発ロードマップを見てもわかるように、Star Suite と OpenOffice.org は、ソースコードの大半を共有している。商用の製品を開発しながら、オープンソースにしている好例である。

Sun Microsystems 社は、オープンソースコミュニティと協調して開発を進めるために、Sun Industry Standards Source License[15] (以下略して SISSL*6)/GNU Lesser General Public License[14]

*6 参考文献 [24] の p.360 によると、SISSL はシズルと読

(以下略して LGPL) のデュアルライセンスを適用することに決定した。

また、世界中に広がっているコミュニティの開発者と協同作業するために、分散開発環境として、バージョン管理システムの CVS を導入した。無制限に書き込みを許可するのではなく、CVS に書き込む (コミットする) 際には、Sun Microsystems 社の Joint Copyright Assignment (以下略して JCA[27]) を許諾、署名している事を条件としている。JCA は、個人の資格での許諾、署名が多いが、法人として許諾、署名している場合もある。

この結果、Sun Microsystems 社によるソースコードの管理は容易になったが、JCA に署名していないプログラマーが OpenOffice.org に改良するコードを書いたとしても反映されずに、各 OS の配布元 (Distributor) に蓄積されている。そのため、OpenOffice.org へのパッチなどで改変内容が異なる場合が出てきている。

また、OpenOffice.org 1.1 の頃は、OpenOffice.org 自前のインストーラでインストールをしていたので、各 Linux Distribution は、独自にパッケージングを行っていた。例えば、

- SuSE Linux[43]
- Mandrake Linux[44]
- Debian GNU/Linux[45]
- FreeBSD[46]

などである、各 Linux Distribution は、個別に導入や運用に関する知識を集め配布のためのパッケージ化を行なっている。

3.2.2 開発ツール概観

現在の OpenOffice.org を開発するのに利用するコンパイラやエディターなどの開発ツールの現状は、Unix の伝統的なスタイルであるコマンドラインからコンピュータに指示するコマンドを入力するスタイルである。いわゆる、Character User Interface (以下、CUI) の “make world” の世界である。

開発に必要なとするソースコードのファイル群は、インターネット上で分散開発が可能な CVS*7 を利用して取得する。

OpenOffice.org は、C/C++ 言語により記述されている。Linux 上でビルドする (実行形式ファイルを構成する) 場合、開発ツールとしてオープンソースのプロダクトである gcc や g++ を利用している。ソースファイル群の中には、Java 言語で記述されたプログラムも多数存在する。このため Java の開発環境も必要である。本論文では Java の環境として、Sun Microsystems のソフトウェア開発キット (Software Development Kit*8 以下 SDK) を使わずに、blackdown.org[47] から入手できる SDK を利用した。

執筆時点*9で、OpenOffice.org をビルドするために、下記のような環境要求がある。

- gtk

む。

*7 Concurrent Version System バージョン管理システム

*8 Java の SDK は、Java のコンパイラとコンパイラが生成するコードを解釈して実行する Java VM (Virtual Machine) などを含む。

*9 2004/08/07

- gtk2+ 2.0.x
 - gcc
 - gcc 3.3
 - Java VM
 - blackdown.org もしくは Sun 純正 Java VM
- 実際に試したのは、上記の Java VM であるが、他の Java VM (IBM / その他の実装系) で構築が可能な場合もある。

この他にも、OpenOffice.org は、多様な環境でコンパイルを行うために dmake という make のパリエーションを使っている。

ここで、OpenOffice.org をコンパイルする環境を整えるために、まず Debian Package 版の OpenOffice.org をビルドするのに必要な環境を整え、その環境を使って Native 版の OpenOffice.org のビルドする環境について話を進める。

著者の開発環境の、Debian GNU/Linux 開発版にある OpenOffice.org をコンパイルするための依存関係は、Debian パッケージの Build-Depends 欄を確認するとわかる。(確認方法は、apt-cache コマンドの showsrc オプションを使用。) Build-Depends 欄に表示されているパッケージだけがインストールされるだけでなく、依存関係を満たすように他に追加的に必要なライブラリなども自動的にインストールされる。

上記の条件を揃えて、Debian Package 版の OpenOffice.org をビルドする。Debian Package 版の OpenOffice.org がビルドできれば、OpenOffice.org の Web サイトより CVS 版のソースコードまたは、tar.gz 形式ソースコードを取得する。

ビルドに必要なリソースは、ディスク容量が約 5G byte である、メモリは多い方が良い、また CPU は速い方が良い、環境の一例としては、Memory 512M byte, CPU Pentium 1G Hz のスペックの PC でビルドを開始してから終了するのに 7 時間程度である。

開発の時には、デバッグ情報の付加、バグの修正などのためにたびたび再コンパイルをおこなうことがある。このため、OpenOffice.org から取得した状態および Debian GNU/Linux の OpenOffice.org 開発版パッケージでは、ccache というソフトウェアを利用して、再コンパイルの高速化を行う。ccache はコンパイラキャッシュで、以前のコンパイル結果をキャッシュし、同じコンパイル処理を再実行されるのを検知して、C/C++ コードの再コンパイル速度を向上させる。

4 開発手法に対する問題

4.1 OpenOffice.org の問題

OpenOffice.org にもオープンソース活動を推進し、多様な開発者の参加を促すにあたって克服すべき問題を以下に示す。

4.1.1 膨大なソースコード

OpenOffice.org のソース構成、プログラムのモジュール構造およびディレクトリー構造としては、OpenOffice.org での個別プロジェクト [31] で分化し、各ディレクトリーに分かれている。

著者らは、OpenOffice.org のプログラムの大きさを測定するために行数を基準にして測定を行った。OpenOffice.org の 1.1.1 版のソースコー

ドを展開して調査した。また、次期 OpenOffice.org 2.0 の β 版 になっている途中のバージョンである SRC640_m65 版を CVS から取り出してプログラムの行数を調査した。

OpenOffice.org 1.1.1 版のソースコードは、約 600 万行であった。(コメントも込み)

次期 OpenOffice.org 2.0 になるブランチである SRC680_m65 版のソースコードでは、約 1,500 万行であった。

拡張子	1.1.1 での行数	SRC680_m65 での行数
cxx	4,511,322	7,714,518
hxx	1,308,130	2,675,342
java	440,730	917,179
py	2,351	1,020,241
c	162,454	1,297,754
h	163,401	1,782,496
小計	6,588,388	15,407,530

表 1 OpenOffice.org 1.1.1 版および SRC680_M65(開発版 Pre 2.0) の言語別ソースコード行数

4.1.2 ビルド環境

OpenOffice.org のような大規模なプログラムでも、個人環境で一連のビルドができるようになったのは喜ばしいことである。しかし、オープンソースプロジェクトとしては、開発者を確保するために開発に参加しやすい環境を作る必要がある。

開発者が開発に参加しやすい環境を作る必要性は、OpenOffice.org のコミュニティとしても意識しており、Web にドキュメントを公開したり、日本のコミュニティ [23] が日本語版をビルドするためにシェルスクリプト [30] を公開している。

しかし、さらなる開発ドキュメントの提供および OpenOffice.org 開発へのハードルの高さを下げる取り組みが行なわれる必要がある。JCA にサインをしている OpenOffice.org の変更権を持つ人および企業数は、文献 [29] によると 404 人^{*10}である。この人数では、世界中での OpenOffice.org のユーザから必要とされる機能を実装し提供することは難しい。

4.1.3 コアの部分が大きい OpenOffice.org

OpenOffice.org は、優れたアプリケーションであるが、アプリケーション毎のデータ埋めこみを容易に実現するためなど、さまざまな理由からコアの部分が大きすぎる構造になっている。このため OpenOffice.org を構成する各ソフトウェアから共通して利用されるライブラリ部分をメモリに読み込んでからでないと起動しない、1 プロセスで構成される規模の大きなアプリケーションになっている。

Unix 版の OpenOffice.org では、共有ライブラリを予めロード (preload) しておくことで起動時

^{*10} 2004 年 8 月調べ

間を短くする仕組みがある。コア部分が大いので起動時間の向上はする。しかし、大幅な改善はしない。

OpenOffice.org を構成する各ソフトウェアから共通して利用されるライブラリ部分が大い。共通部分が大いのは、OpenOffice.org の内部に Object Broker の仕組みである UNO[33] 機能を実装されており、UNO が Object Broker のサービスを提供するのに関連するオブジェクトを要求するため、UNO に関連する部分が大きくなるからである。

5 IDE と Eclipse

一般的に、統合開発環境 (IDE) といえば、パーソナルコンピュータの分野ならば、Microsoft 社の Visual Studio[34] や Boland 社 [35] の Turbo C などの Turbo シリーズ、Delphi などがある。また、オープンソースで開発されている IDE としては、Eclipse、Code Crusader[36] や Wide Studio[37] などが存在する。現在の所、オープンソースソフトウェアの中でも活動の頻度の高さ、およびプラグインの数から Eclipse が先んじていると考えている。

本章では、5.1 節にて、IDE の長所と短所について述べ、5.2 節にて、オープンソースの IDE である Eclipse について述べる。

5.1 IDE の長所と短所

既存の CUI 環境は、歴史的経緯で Unix では計算機に要求するリソースは少ないほど良いという思想があり今でも受けつがれている。

しかし、現在では、18 ヶ月で CPU の性能が 2 倍になるといふ、Moore の法則におおまかに沿う形で CPU Power が増大、ビット単価 — メモリ容量およびハードディスク容量 — の拡大が起きており、個人が所有する計算機のリソースが飛躍的に拡大している。

増える計算機リソースを、どのような用途に利用するのか。という問いに対して、利用者、とくにプログラマへの支援を行なうというアプローチが IDE である。

IDE は、プログラマのプログラミング支援に特化したプログラムである。IDE はプログラムによって多少の違いはあるが、ソースコードデバッグ、プログラムのビジュアル化によるプログラマの「気づき」への支援、プログラム入力支援などの機能を持っている。

IDE の長所と短所は、表裏一体になっている。計算機資源を、人間がプログラミングを行う事に対するサポートに費しているためであり、以下のようになる。

- 長所
 - 開発者の利便性が向上する。
 - 開発者の開発に対するハードルが下る。
- 短所
 - 計算機資源を要求する。

計算機資源が増大している現在においては、IDE を利用することは人間がプログラミングを行う事に対するサポートを実現する上で十分にリーズナブルであると考えられる。

5.2 Eclipse とは

Eclipse はオープンソースで提供される統合開発環境である。文献 [10] によると、もともと IBM 社の Visual Age シリーズを置き換える次世代統合開

発環境として、40 名の専属開発者によって開発がスタートした。オープンソースであるので入手も容易であり、Java の開発環境としてメジャーな位置にある。

Eclipse は、プラグインを導入することにより Java 以外の言語もサポートすることができる。たとえば、Eclipse 用の C/C++ を使った統合開発環境である CDT[39] (図 3) やスクリプト言語 Ruby 統合開発環境である rubyclipse [40] などの多様な開発環境が存在する。

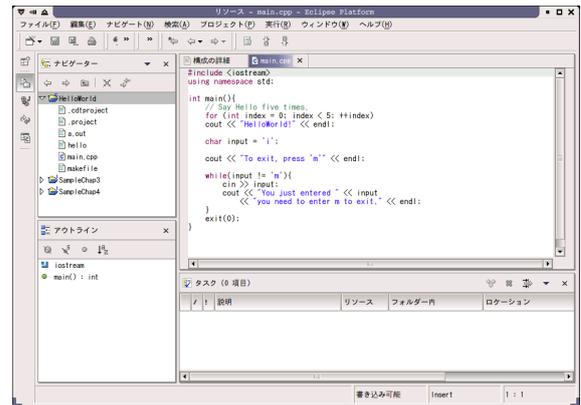


図 3 Eclipse 2.1.3 CDT 1.2.1 (C/C++ 開発環境)

それぞれの言語のプラグインによって詳細は異なるが、Eclipse は優れたプログラミング支援機能を持っている。コードテンプレート機能を使うと、プログラミングで良く使うフレーズを登録できる。

オブジェクト指向言語であれば、クラスのブラウジングなども可能になっている。オブジェクトの階層に沿ってメソッド名など補完を行うこともできる。

6 Eclipse による OpenOffice.org の開発

本章では、Eclipse を利用して OpenOffice.org の開発を行うための手法およびノウハウについて論述する。Eclipse に前述の CDT を導入して、OpenOffice.org の大部分を占めている C/C++ および Java について容易に開発を行う環境を整える。

6.1 Eclipse を使う利点

オープンソースな統合開発環境である Eclipse で OpenOffice.org の開発を行うと

- Eclipse の持つ、テンプレート機能やメソッド補完機能などのプログラミング支援機能を利用することができる。
- ソースコードの Visual 化などが可能になり、プログラマのプログラミングへの参加の障壁を下がる。
- Eclipse は、プロジェクトの中に入れ子でプロジェクトを作ることができる。このためモジュールの整理がやりやすくなる
- 既存の OpenOffice.org の開発で利用されている dmake を使った手法よりもプログラミングの敷居が低くなり、プログラマの参加がしやすくなる。

ことが期待できる。

項目	細目	特記
ハードウェア	メモリ 512M byte CPU 1G Hz HDD 40G byte	
ソフトウェア	OS:Debian GNU/Linux Eclipse 2.1.3 CDT 1.2.1 RubyEclipse(rdt) 0.3.2	開発版を利用

表 2 Eclipse から OpenOffice.org をビルドするスペックの例

6.2 Eclipse 開発環境

現在、開発環境として利用しているのは、表 2 の環境である。

表 2 上記の構成では、メモリ不足であり、1G byte 以上のメモリが望ましい。

6.3 Eclipse による OpenOffice.org のビルド方法

本節では、Eclipse を用いて OpenOffice.org のビルドを行う一連の手順について述べる。ここで利用しているシステムは、Debian GNU/Linux の開発版である Debian GNU/Linux unstable を利用している。そのシステム上に、blackdown 版 JAVA をインストールして、Eclipse などをインストールしている。またターゲットとしては、Debian Project がメンテナンスしているパッケージの OpenOffice.org ではなく、OpenOffice.org プロジェクト [22] で配布しているソースコードとする。

Eclipse の C/C++ プラグインである CDT をインストールする。2 通りの方法が考えられる。Eclipse のプラグインマネージメント機能を利用する。もう一つは、Debian 流のパッケージングシステムを利用する方法がある。いずれにしても、CDT のインストール方法については、CDT[39]にある。FAQ、チュートリアルや User's Manual なども充実している。著者は、Eclipse の CDT を Debian パッケージを作成して、Debian GNU/Linux システムから管理しやすい形を選択した。

Eclipse より、C/C++ プロジェクトを生成する。OpenOffice.org のソースコードをインポートする。(図 4 および、図 5) このソースコードをインポートすると時間は、約 10 分ほどかかる。これは、OpenOffice.org のソースコードが巨大であるため、Eclipse がソースコードを読み取りつつ、プロジェクトを生成しているため作業してる PC の上で主にメモリのリソースが不足しているからである。苦労した点は、初期の段階ではメモリ不足になってインポートができなかったことである。ターゲットマ

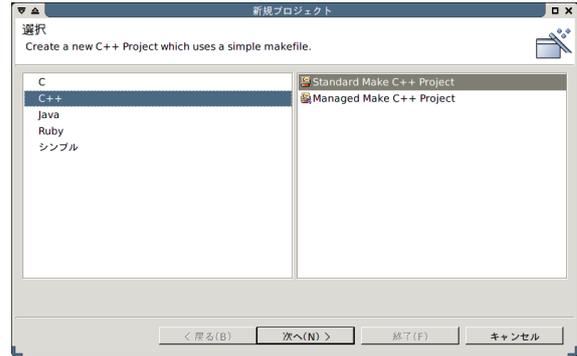


図 4 Eclipse における CDT のプロジェクト追加図 1

シン上の Debian GNU/Linux にて、不要なサービスやアプリケーションを停止したりなど、Eclipse のメモリ割り当てに使える量を増加させた。オプションに、Eclipse のオプションとして “-vmargs -Xmx512M” を指定することで OpenOffice.org のソースコードがインポート可能になった。

事前に 3.2.2 節にあるように、下記のようなコマンドを用いて、事前に Debian Project の OpenOffice.org パッケージを再ビルドできる環境を作成しておく。

```
apt-get source openoffice.org
```

として、ソースコードを取得して debuild コマンドでビルドする。ビルドに必要な開発ライブラリで不足している開発ライブラリを列挙する機能がある。また、apt-get コマンドに build-dep というオプションがあり自動で開発ライブラリなどを導入することが可能だ。このように Debian Project 版の OpenOffice.org をビルドする環境を整えて、OpenOffice.org 本家版の作業に移る。

その後、OpenOffice.org コミュニティが作成した CUI 用の環境設定スクリプトである jaconfigure.sh を実行してコンパイル環境を整える。コンパイル環境を整えた後に、LinuxIntelEnv.Set.sh というシェルスクリプトが生成されているので、これを雛型にして、コンパイル用のシェルスクリプトを生成する。具体的には付録??にある例のようにシェルスクリプトの最後に dmake を呼出すようにして、Eclipse からは、シェルスクリプトの呼び出し (図 6) としておき、オプションに、Eclipse のオプションとして -vmargs -Xmx512M として、このメモリ環境でもコンパイルできた。

6.4 問題点

- Eclipse の使用方法の作法に倣う必要がある。これは Eclipse への慣れの問題でもある。また Eclipse は操作をカスタマイズ可能であり、この問題を軽減できる。
- PC のリソースを IDE が大量消費する。技術的な問題ではない。PC のリソース、能力は増大の一步を辿っているため時間が解決する問題である。

7 ooEditor の開発

本論文で提案する ooEditor とは、OpenOffice.org の ooWriter をもとに作成し、主にテキ

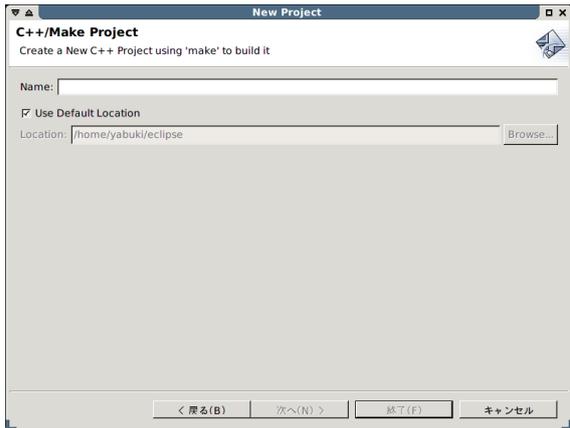


図 5 Eclipse における CDT のプロジェクト追加図 2

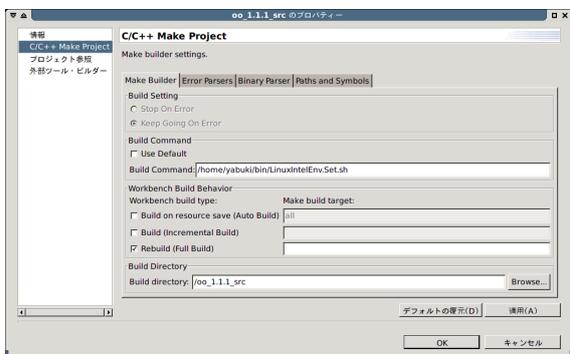


図 6 Eclipse へ OpenOffice.org のビルドスクリプトを登録

ストエディタの機能を持つプログラムである。このプログラム作成には、いくつかの目的がある、(1) 最小の動作セットを作成して、OpenOffice.org の実行時のモジュール分割を提案する。つまり必要な時に必要なモジュールをロードすることでリソースの効率的な利用を提案する。(2) 主にテキストを扱う人々のために軽量な OpenOffice.org を提供する。(3) 初等教育用途の OpenOffice.org のサブセット版を作成することにより、初等情報教育の場で OpenOffice.org の利用を通してオープンソースを学ぶ機会を作る。本論文では、プロトタイプを示して、OpenOffice.org コミュニティへ提案書を提出する。

7.1 ooEditor 開発の意義

4.1.3 節にて議論したが、OpenOffice.org は、コアの部分が大きい構造になっている。この構造は、ooWriter の中に ooCalc (表計算) のデータや ooDraw のデータを埋めこんで処理するには便利な構造である。OpenOffice.org は、巨大な単一バイナリのソフトウェアである。より使いやすいソフトウェアとする一つの方法は、必要なコンピュータ資源を少なくすることである。現在の大きなコアの部分をメモリに読みこまなくてはならない構造から、小さいコアで利用者が必要な機能をプラグイン化して動的に必要なプラグイン

を組み込む構造にする方法が有効であると考えられる。上記の理由により、最小の動作セットとして ooEditor を作成する。ooEditor を使えば、いろいろな人々の「最小限」の要求を満たすことができる。最小限の機能では足りない場合は、プラグインを利用して機能を拡張する。利用シーンにあわせて、プラグインを抜き差しする方式として提案する。オープンソースソフトウェアの場合、具体的に動いている実例を見せると海外の開発者も動きやプログラムから興味を持って開発に参加してもらえる。このような観点から ooEditor の仕様としてプロトタイププログラムを作成する。最小の動作セットは、初等教育の場においても有益だと考える。初等教育の児童が、OpenOffice.org を利用するとき、最小の動作セットの基本機能で操作可能であることは、その選択肢の少なさから、児童の迷う余地を減らす。

また、最小の動作セットは、ooEditor が要求する実行環境のスペックを低くても動作することが可能である。比較的性能が低いパーソナルコンピュータであっても動作させることが可能であることは、ooEditor が利用できる環境が広がることを示している。また、OpenOffice.org と同じスペックならば ooEditor の方がプログラムの動作が軽快に動作することが予測できる。実行環境のスペックを押えるということは、国の e-japan 政策 [21] の「(2) 学校教育の情報化等」の部分で一人一台の環境整備の部分において利用できるコンピュータの幅を広げることができる。

また、ooEditor は、プラグイン形式で拡張可能である。このため、児童の ooEditor への習熟度合によっては、プラグインを導入して、より高度な利用を望む児童の要求を満たすことも可能である。初等教育の児童が、ooEditor を利用して、ooEditor、OpenOffice.org やもっと幅広くいうとオープンソースソフトウェアに興味を持つことで、将来の OpenOffice.org 開発者やオープンソースソフトウェア開発者となる可能性がある。

7.2 ooEditor とは

7.1 節にて、ooEditor の開発の意義について述べた。本節では、ooEditor 自体について述べる。ooEditor の機能として、3.1.3 項にある、OpenOffice.org の ooWriter の機能から最小限度の機能を抜き出しこれを定める。この最小限度の機能に入らなかった機能はプラグインとして、実装される。

OpenOffice.org の ooWriter を機能過剰だと思う人には、ooEditor を使って利用者の必要に応じたソフトウェアにカスタマイズが可能である。

以下の 7.3 節で、小学生の利用や最低限のビジネス文書で必要最小限の部分を検討する。

次に 7.4 節で、ooEditor の仕組について概略の説明を行い、それを受けて 7.5 節の提案を行なう。

7.3 最小の機能とは

著者は、教育系の情報化を研究している院生、ビジネス系の IT リテラシーを研究している院生および指導教官との間で議論*11をおこなった。この議論の中から、教育現場やビジネス文書の最小限必要とされる機能を検討した。ooEditor をワードプロセッサ風の機能から Editor 風の機能に絞り込むこと方針で ooEditor に必要な機能を検討した。

*11 とりわけ、指導教官である中野秀男教授、和田和子さんと孫 初美さん

表 3 がその検討結果である。

項目	細目	特記
文字入力		
ファイル操作	Load Save	
文字揃え	右揃え 中央揃え 右揃え	
フォント操作	フォントの指定 フォントのポイント指定	
(選択範囲の) 文字色変更		
Undo 機能		
検索機能		検索された 対象が反転する。
置換機能		
行番号表示機能		
ツール・バー		ツールバーには、 少くともファイルの save についてアイコンが存在すること
印刷機能		

表 3 ooEditor における、最小の機能表

7.4 ooEditor 仕組の概観

ここでは、ooEditor の仕組の基本概略を説明する。

ooEditor は、Text Editor のように GUI 環境下で、非同期で命令を受け付けながら処理を進めていける仕組が必要である。Model, View, Controller の 3 つの独立した処理から、GUI を構成する MVC model[49] を意識しつつ、無限実行可能な Undo を実装するために、GUI からの指令は、いったん queue に入れて queue の read pointer と write pointer を管理する。(図 7)

通常の場合は、queue に入っている GUI からの

イベント (ex. 文字が入力された。文字のフォントを変更した。など) を、write pointer が管理をして queue の中に入れていく。

View に関連する処理を行う部分では、Read pointer からイベント・データを読み出し、write pointer と等しくなるまで Event の Dequeue 処理を行ない画面の表示の更新を行う。

実際に、undo の操作が行われた時には、queue の最後に入っている GUI からのイベント (命令) を、無効にしてゆく。Read pointer, write pointer も queue の中に前後する。この方法であれば redo も、write pointer の位置を変えていくだけで可能になる。

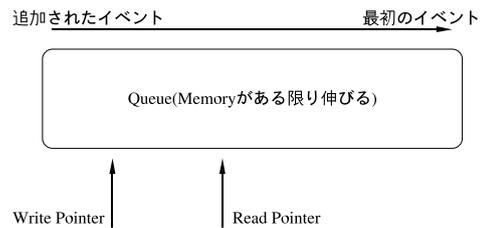


図 7 ooEditor undo queue image

この queue が大きくなると、メモリを圧迫するようになるとすると、file への save 時に、queue を初期化する方法がある。

イベントを queue から取り出して Event の分岐で処理を行なう。メニューの選択や選択範囲の文字種変更などは、イベントに変換される。queue を通して MVC の M つまり、Model の部分でこれらのイベントを処理してゆく。(図 8)

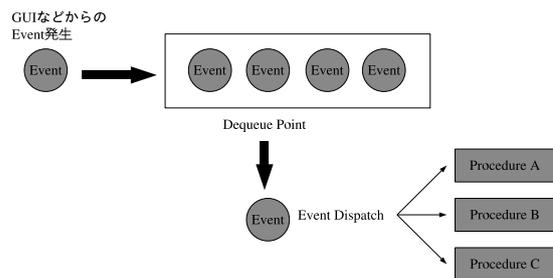


図 8 ooEditor Event Structure

7.5 ooEditor のプラグイン機能

利用者が必要とする機能を組み込み、必要なりソースだけを消費するプラグインという仕組を提案する。

OpenOffice.org の最小セットとして 前節までに ooEditor を提案した。このプラグイン機能は、

ooEditor で最小限の機能を提供して、次のステップとして ooEditor に「機能拡張」を行う仕組みとして位置付ける。

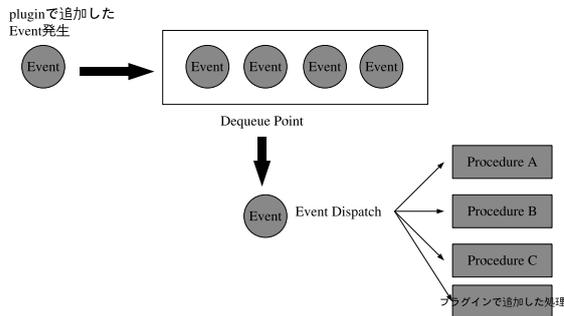


図9 ooEditor プラグイン処理追加

このプラグインによる機能拡張の仕組みは、図9のようになる。

このプラグインの基本的なアイデアは、Dispatch されてきた Event の分岐先を増やすことである。そのために、プラグインでは ooEditor のメニューに登録を追加するなどの Event 発生フックを追加する。

つぎに、プラグインとして機能を拡張するために、依存関係とバージョンの導入を提案する。ooEditor がプラグインを組込む時に、正しく動作することを保証する手段である。ある機能が存在することを保証したり、プラグイン間の調停を行うものである。

プラグイン間の依存関係については、Debian GNU/Linux のパッケージ管理の部分を参考 [60][61] にして下記の表4の内容で提案する。

依存関係名	意味
依存 (Depend)	このプラグインが動作するのに、必要な ooEditor のバージョンと、他のプラグインの情報を記述する。
衝突 (Conflict)	プラグイン同士が排他的な関係であるときに設定する。
置換 (Replace)	既存のプラグインを置き換える状態

表4 ooEditor およびプラグイン間での依存関係記述の提案

この依存関係の維持に関しては、ooEditor とプラグインおよびプラグイン間で連携する必要がある。

7.6 ooEditor プロトタイプ作成

7.6.1 開発環境

ooEditor のプロトタイプは幅広い人から意見を集めることを可能にするために、さまざまなプラットフォームで稼働させることを考える。また、プラグイン形式で、機能を拡張可能にするために、動的結

合が利用可能な言語が望ましい。

OpenOffice.org のデータは、Zip 形式で圧縮された XML 形式のファイルである。このため XML 形式のファイルを取りあつかうライブラリーが存在する事が望ましい。また、XML の中身は UTF-8 であり必要に応じて表示ロケールの文字コードを変換可能である事が望ましい。

GUI に関しては、可能な限り Plathome independent が望ましい。これを考慮し、Unix (Linux, Mac OS X) や Windows で利用可能な Window Toolkit として Tcl/Tk の Tk を利用する。

ユーザインターフェースのメッセージを国際化するために、指定される Locale に応じてメッセージの表示が可能な gettext の枠組を利用する。

上記の要件を満たして、プログラミングを行なうために下記の枠組を利用する。

項目	細目	特記
プログラミング言語	Ruby[50]	
GUI Toolkit	TK Ruby/TK[52, pp. 279-397][55]	
XML ライブラリー	rexml[56][57]	Ruby 1.8 に附属
ユーザインターフェース (UI) の国際化	gettext — Ruby Gettext[58]	
UTF-8 を含む漢字コード変換	ruby iconv[59]	Ruby 附属の nkf ライブラリーでは、UTF-8 を変換できないので、追加の変換ライブラリーを導入する。

表5 ooEditor 作成で利用するツール・ライブラリー

分散開発環境として、CVS を使っている。図10では、開発中の ooEditor の GUI 部についてのイメージ図である。

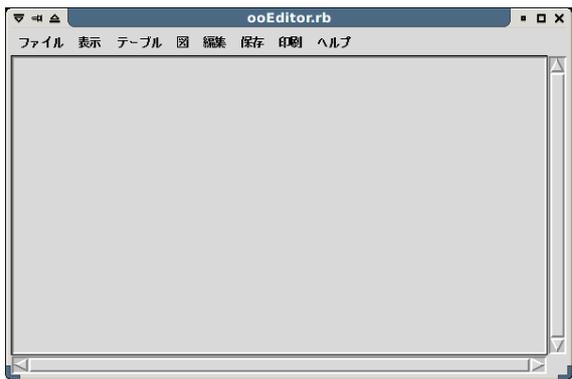


図 10 ooEditor

7.6.2 プロトタイプ作成の概要

生産性の高いスクリプト言語を利用し、GUIの部分は高レベルの Window Widget を利用し作業効率を高める。また高レベル Widget を利用するため、テキストの入力 (XIM を使って日本語も入力可能) やスクロールも容易に実装可能となる。OpenOffice.org の Native Format の `sxw` 形式は Zip 形式で圧縮された XML 形式のファイルである。また、OpenOffice.org で利用されている XML 形式のファイルは、UTF-8 エンコードである。このため、テキストデータを読み取る部分は、REXML ライブラリを使って、テキストデータを取り出しコンピュータのシステムエンコード (著者の環境では EUC-JP) に変換することとする。さらに、GUI のメニュー部分などのメッセージカタログの部分に関しては、`gettext` ライブラリを用いて日本語を表示させることで実現できる。

2005 年 1 月 30 日現在、プロトタイプは完成していない。生産性の高いスクリプト言語を利用し、REXML ライブラリ以外は調査を完了しているので、作業する時間を確保すれば完成する予定である。

個々の進捗としては、GUI の部分は、高レベルの Window Widget を利用しており、テキストの入力 (XIM を使って日本語も入力可能) やスクロールは可能である。`sxw` 形式を読み取って内部のデータを読み取って、システムエンコードへ変換している、変換したデータは REXML ライブラリを用いて個々の XML 要素を操作を行えるようになっている。また `gettext` ライブラリでロケールに応じてメッセージカタログを切り替えることも可能である。EUC-JP 環境で日本語の表示もできている。

8 結び

本論文では、OpenOffice.org の開発を行うにあたり、既存のコマンドラインの開発環境と、オープンソースの統合開発環境として脚光を浴びている Eclipse を比較した。次に OpenOffice.org の開発に Eclipse を利用する提案を行い、Eclipse から、OpenOffice.org をビルドする方法について述べた。

更に、`ooEditor` として OpenOffice.org の Writer の部分について最小限機能セットを検討して提示した。`ooEditor` が持つ意義は、利用者が要求する機能とプログラムが要求するコンピュタリソースとの間で最適な地点を設定できることである。

`ooEditor` は、基本部分からプラグインによって

拡張可能である。このため初等教育の場面において、児童が基本的な操作から、児童の要求に応じてステップアップすることができる。児童の中には、`ooEditor` に触れて、内部を理解したいと望めば、`ooEditor` は、オープンソースソフトウェアであるので内部を見ることができ、`ooEditor` で学んだ児童から、オープンソースにて既存の優れたソフトウェアに触れて、多くの優れたソフトウェア技術者が生れる可能性がある。

本論文における問題意識は、オープンソースソフトウェア開発者の数を増して、オープンソースソフトウェアのすそ野を広げ、頂上を押し上げることである。本論文を書き上げたことにより、その一助ができたと考えている。

本論文での成果は、著者がオープンソースコミュニティへ提案する。このような活動もオープンソースソフトウェアに対する貢献になっていると考えている。

謝辞

本論文を書く上で、多数の方にご支援、ご協力いただき謝辞を述べます。

特に御指導いただきました、大阪市立大学創造都市研究科の中野秀男教授、大西克実講師に感謝いたします。

論文を書く上で `ooEditor` が持つべき最小限の機能について議論をさせていただいた、大阪市立大学大学院 創造都市研究科 都市情報学専攻基盤研究分野の孫初美氏、和田和子氏に感謝します。また研究する上で、研究内容は違えどもアプローチの方法などで、井川裕基氏、上田達也氏、小松久美子氏、柳原秀基氏、李ケイ氏をはじめ、大阪市立大学 大学院 創造都市研究科 都市情報学専攻 基盤研究分野のみなさんにお世話になりました。

ソフトウェアシンポジウム 2004(SS2004)にて、叱咤激励していただいた皆様、とりわけ北陸先端大学の藤枝先生、東田雅宏氏 (PFU) に感謝します。

Debian GNU/Linux でのビルドについて色々とお相談させていただきました。Nakata Maho 氏 Naohiko Saito <hu3n-situ@asahi-net.or.jp> 氏と、OpenOffice.org ユーザ会の森沢崇幸 (ikuya@good-day.co.jp) 氏に感謝します。

この研究を行うにあたり、オープンソースコミュニティの方々にお世話になりました。

最後に、著者を支援して、著者を働きながら大阪市立大学 大学院へ進学することを許可し、本研究をさせて頂いている (株) グッデイに感謝します。

参考文献

- [1] Eric Raymond (著) 山形浩生 (翻訳) 「伽藍とバザール: オープンソース・ソフト Linux マニフェスト」, 光芒社, 1999 年 9 月
- [2] <http://cruel.org/freeware/cathedral.pdf>
The Cathedral and the Bazaar 翻訳版 (2004/08/06 確認)
- [3] <http://cruel.org/freeware/noosphere.html>
第 2 部 「ノウアスフィアの開墾」(2004/08/06 確認)
- [4] <http://cruel.org/freeware/magicpot.html>

- 第3部「魔法のおなべ」(2004/08/06 確認)
- [5] Richard M. Stallman (著者) (株) ロングテール/長尾 高弘「フリーソフトウェアと自由な社会」 - Richard M. Stallman エッセイ集, (2003/05)
- [6] グリン ムーディ (著) 小山 裕司 (翻訳) 「ソースコードの反逆— Linux 開発の軌跡とオープンソース革命」, (2002/06)
- [7] Jan Sandred 著 でびあんぐる 監訳 「オープンソースプロジェクトの管理と運営」, 2001,
- [8] WONDERVIEW PRODUCTIONS presents JTS ムーア FILM 「レボリューション OS」, (2003/08/22)
- [9] <http://www.vasoftware.com/>
VA SOFTWARE (2005/01)
- [10] OSPG Eclipse SIG 「開発のプロが教える 標準 Eclipse 完全解説」, (2003/08)
- [11] http://www.opensource.org/docs/definition_plain.html
The Open Source Definition (2004/12/30 確認)
- [12] <http://www.opensource.jp/osd/osd-japanese.html>
オープンソースの定義 (2004/07/27 確認)
- [13] <http://www.opensource.jp/gpl/gpl.ja.html>
GNU 一般公衆利用許諾契約書 (2004/01/30 確認)
- [14] <http://www.gnu.org/copyleft/lesser.ja.html>
GNU 劣等一般公衆利用許諾契約書 (2004/08/07 確認)
- [15] http://ja.openoffice.org/sissl-ja_01.html
Sun Industry Standards Source License Version 1.1 SISSL 日本語訳 (参考) (2004/08/07 確認)
- [16] 秋本 芳伸 (著), 岡田 泰子 (著) 「オープンソースを理解する」, ディーアート, (2004/01)
- [17] 米持 幸寿 (著), 日経 BP Java プロジェクト (編集) 「オープンソースがビジネスになる理由 — 勝ち組企業は何をしたか」, 日経 BP 社, (2003/10)
- [18] 川崎 和哉 (著) 「オープンソースワールド」, 翔泳社 ; (1999/12)
- [19] 末松 千尋 (著) 「オープンソースと次世代 IT 戦略 — 価格ゼロ時代のビジネスモデル」, 日本経済新聞社, (2004/05)
- [20] クリス ディボナ (著), マーク ストーン (著), サムオックマン (著), Chris DiBona (原著), Mark Stone (原著), Sam Ockman (原著), 倉骨 彰 (翻訳) 「オープンソースソフトウェア — 彼らはいかにしてビジネススタンダードになったのか」, オライリー・ジャパン, (1999/07)
- [21] <http://www.kantei.go.jp/jp/singi/it2/kettei/ejapan2004/040615-32.html>
II 重点政策 5 分野 2. 人材の育成並びに教育及び学習の振興 (2005/01)
- [22] <http://www.OpenOffice.org/>
OpenOffice.org WebPage (2004/02/19 確認)
- [23] <http://ja.openoffice.org/>
OpenOffice.org 日本ユーザー会 (2004/07/29 確認)
- [24] OpenOffice.org 日本ユーザー会 (著) 「OpenOffice.org オフィシャルユーザーズガイド」, (2004/05)
- [25] OpenOffice.org 日本ユーザー会 (著) 「OpenOffice.org オフィシャルユーザーズガイド」, 2 章, 3 章 (2004/05)
- [26] http://porting.openoffice.org/porting_overview.html
OpenOffice.org での porting 状況 (2004/08/06 確認)
- [27] <http://openoffice-docj.sourceforge.jp/tr/translated/jca.html>
Joint Copyright Assignment (JCA) の参考訳 (2004/07/27 確認)
- [28] http://oooug.jp/mirror/documents/QandA/tr_most.html#7
OpenOffice.org FAQ — StarOffice と OpenOffice.org の違い? (2004/07/28 確認)
- [29] <http://www.openoffice.org/copyright/copyright-approved.html>
Copyright-Approved Persons and Companies (2004/08/10 確認)
- [30] http://ja.openoffice.org/documents/develop/build_linux.html
Linux でビルドする方法 (2004/08/07 確認)
- [31] <http://projects.openoffice.org/accepted.html>
Accepted Projects Category (2004/07/28 確認)
- [32] <http://ccache.samba.org/>
ccache (2004/02/22 確認)
- [33] <http://udk.openoffice.org/>
UNO Development Kit (UDK) project (2005/01/23)
- [34] <http://www.microsoft.com/japan/msdn/vstudio/default.aspx>
Visual Studio ホームページ (2004/08/10 確認)
- [35] <http://www.borland.com/>
Borland: Leading Provider of Technology for Software Applications (2004/08/10 確認)
- [36] <http://www.newplanetsoftware.com/jcc/>
Code Crusader (2004/08/10 確認)
- [37] <http://www.widestudio.org/index.html>
WideStudio Home page (2004/08/10 確認)
- [38] <http://www.eclipse.org/>
eclipse.org (2004/02/19 確認)
- [39] <http://www.eclipse.org/cdt/>
CDT — C/C++ Development Tools (2004/02/22 確認)
- [40] <http://sourceforge.net/projects/rubyecclipse>
rubyecclipse (2004/02/22 確認)
- [41] <http://sourceforge.net/projects/trustudio/>
trustudio-python (2004/02/24 確認)
- [42] <http://sourceforge.net/projects/pyeclipse/>

- Python Plugin for the Eclipse platform
(2004/02/24 確認)
- [43] <http://www.novell.com/linux/suse/>
Novell SUSE LINUX (2005/01/23)
- [44] <http://www.mandrakelinux.com/ja/>
Mandrake Linux (2005/01/23)
- [45] <http://www.jp.debian.org/>
Debian GNU/Linux (2005/01/23)
- [46] <http://www.jp.freebsd.org/>
The FreeBSD Project (Japan) (2005/01/23)
- [47] <http://www.blackdown.org/>
Java Linux (2005/01)
- [48] <http://www.ogis-ri.co.jp/otc/hiroba/technical/Squeak5/S5-3-1.html>
オブジェクトの広場 Happy Squeaking!!
(2005/01)
- [49] <http://www.atmarkit.co.jp/fjava/javafaq/j2ee/j2e07.html>
MVC モデルという言葉をよく聞きますが何のことですか? (2005/01)
- [50] <http://www.ruby-lang.org/ja/>
オブジェクト指向スクリプト言語 Ruby
(2005/01)
- [51] 高橋 征義 + 後藤 裕蔵 著, まつもろ ゆきひろ 監修, 「たのしい Ruby - Ruby ではじめる気軽なプログラミング」(ソフトバンク パブリッシング 株式会社, 2002 年) ISBN4-7973-1408-7
- [52] 前田 修吾 まつもと ゆきひろ やまだ あきら 永井 秀利 共著, 「Ruby アプリケーションプログラミング」(オーム社, 2002 年) ISBN4-274-06461-1
- [53] まつもと ゆきひろ 著 「Ruby デスクトップリファレンス」(オライリー・ジャパン/オーム社) ISBN4-87311-023-8
- [54] デビット・トーマス + アンドリュー・ハント = 著 田和 勝 = 訳 まつもろ ゆきひろ = 監修 「プログラミング Ruby - 達人プログラマーガイド」(ピアソン・エデュケーション, 2001 年) ISBN4-89471-453-1
- [55] 永井 秀利 (著) 「Ruby を 256 倍使うための本界道編」, アスキー, (2001/12)
- [56] <http://www.germane-software.com/software/rexml/> REXML - Home
(2005/01)
- [57] 須藤 功平 (著) 「Ruby de XML - Ruby で始める XML!」, オーム社, (2002/11)
- [58] <http://ponx.s5.xrea.com/hiki/ja/ruby-gettext.html>
Ruby-GetText-Package (2005/01)
- [59] <http://www.ruby-lang.org/ja/man/index.cgi?cmd=view;name=Iconv> Iconv
(2005/01)
- [60] <http://www.debian.org/doc/debian-policy/ch-relationships.html>
Chapter 7 - Declaring relationships between packages (2005/01)
- [61] <http://qref.sourceforge.net/Debian/reference/ch-system.ja.html#s-depends>
Debian リファレンス - Debian の基礎知識 — 2.2.8 パッケージの依存性 (2005/01)
- [62] <http://www.icot.or.jp/FTS/REPORTS/>
H14-reports/H1503-AITEC-Report4/AITEC0203R5-html/AITEC0203R5-ch4-2.htm
米国の連邦政府 R&D 計画における省庁間の役割分担と連携の仕組み (2005/02)