Open for Change

Scoping Paper on the Use of FLOSS in Cadastre and Land Registration Applications

Prepared by
Gertrude Pieper Espada

October 2007

FAO Land Tenure and Management Unit (NRLA)
FIG Commission 7
World Bank Thematic Group on Land Administration

ABBREVIATIONS

BSD	Berkely Software Distribution
CCDM	Core Cadastral Domain Model
COTS	Commercial Off-The-shelf Software
ESRI	Environmental Systems Research Institute
FAO	Food and Agriculture Organisation
FIG	International Federation of Surveyors
FLOSS	Free / Libre Open Source Software
FOSS4G	Free and Open Source Software for Geoinformatics
FSF	Free Software Foundation
GIS	Geographic Information System
GPS	Global Positioning System
GML	Geography Markup Language
GNU GPL	GNU General Public License
GNU LGPL	GNU Lesser General Public License
GRASS	Geographic Resources Analysis Support System
GUI	Graphic User Interface
gvSIG	Generalidad Valencia Sistema de Información Geográfica
IDABC	Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens
IT	Information Technology
ILWIS	Integrated Land and Water Information System
INPE	Brazilian National Space Institute
ITC	International Institute for Geo-Information Science and Earth Observation
JUMP	JAVA Unified Mapping Platform
KDE	K Desktop Environment
LAN	Local Area Network
MBR	Minimum Bounding Rectangle
MEH	Ministerio de Economía y Hacienda (Spain)
MERIT	Maastricht Economic and social Research and training centre on Innovation and Technology
NRLA	FAO Land Tenure and Management Unit
OGC	Open Geospatial Consortium
OGF	Open GRASS Foundation
OSGEO	Open Source Geospatial Foundation
OSS	Open Source Software
PDA	Personal Digital Assistant
QGIS	Quantum GIS
RDBMS	Relational Database Management System
SDBMS	Spatial Database Management System
SFS	Simple Feature Specification
SPRING	Sistema de Processamento de Informações Georeferenciadas
SQL	Structured Query Language
uDIG	User-friendly Desktop Internet GIS
UML	Unified Modelling Language
WAN	Wide Area Network
WFS	Web Feature Service
WFS-T	Transactional Web Feature Service
WMS	Web Map Service

SUMMARY

The open source movement has attracted world-wide attention, and open source software is increasingly used as an alternative to proprietary software products. Yet, there is little evidence of the use of open source software in digital land administration systems.

FAO Land Tenure and Management Unit, FIG and the World Bank Thematic Group on Land Administration support the development of sustainable and affordable systems for land tenure security and land administration. In this context, the use of Free / Libre and Open Source Software (FLOSS) for cadastre and land registration systems is reviewed.

First, the concepts of FLOSS in general and FLOSS for geospatial applications are discussed, as well as the functioning of open source developer communities. Then, land administration systems and their software requirements are analysed, and an overview is given of available open source alternatives. The core of digital cadastre systems generally consists of a data repository containing cadastral boundaries and land register data. When comparing FLOSS database products PostgreSQL and MySQL with proprietary database software (Oracle), it is found that Oracle performs better in most aspects, but the FLOSS products are developing into valid alternatives. As repository for spatial data, PostgreSQL with PostGIS comes closer to the sophistication of Oracle Spatial than MySQL.

GIS software is needed for the manipulation and updating of cadastral map data. A review of a number of FLOSS desktop GIS software products revealed that while map editing functionality is still limited in FLOSS tools, some have good potential to be used in cadastral systems. Several of the reviewed FLOSS GIS products have the advantages of platform independency and interoperability through the compliance with open standards, good topology validation tools, and multiple interface translations which makes these products especially useful in non-Anglophone countries.

Apart from GIS software, cadastral and surveying applications are used to maintain a cadastral fabric and facilitate typical cadastral functions such as consolidations and subdivisions of land parcels. While several commercial cadastral and surveying software products are available as extensions for GIS software, no FLOSS alternatives are found. Since the available proprietary cadastral and surveying software cannot be combined with FLOSS GIS software, the needed cadastral functionality must be developed for FLOSS GIS if it is to be used in a cadastre system.

Developing countries as well as economically more advanced countries can benefit from the use of FLOSS, all be it in different ways. The study finds that the use of FLOSS for cadastre and land registration systems in developing countries would be most successful in a context where governments actively support the use and development of open source software as part of their national IT strategy. In such a context, local IT companies are stimulated to develop innovative FLOSS-based solutions and could take part in the development of digital cadastre and land registration systems. In economically advanced countries, the use of FLOSS is most evident in online information systems and internet mapping. A series of internet specifications from the Open Geospatial Consortium (OGC) and the development of open source WebGIS applications have opened the way for the development of cost-efficient online cadastral enquiry systems. New standards such as the Geography Markup Language (GML) and Web Feature Service (WFS) which go particularly well with open source

software could radically change the way in which cadastral datasets are maintained and updated within the next couple of years.

FAO and partners can contribute to the development of sustainable and cost effective land administration systems by supporting the use of FLOSS. When planning for the development of digital systems in land administration projects in developing countries, the customization of FLOSS products by local IT companies can be considered as an alternative to procurement of foreign proprietary software solutions. While this is not easy, the use of FLOSS for cadastre and land registration systems could be supported further by the development of open source cadastral extension software for an existing FLOSS GIS product. An important step has already been made through the development of the Core Cadastral Domain Model (CCDM) which sets a standard for cadastral data and functionality and forms a guideline for the development of digital land administration systems. While it is unrealistic to expect that the CCDM will fit every cadastre and land registration system in the world, adopting the CCDM standard will give the advantage of interoperability and international cooperation in the development and functioning of cadastre systems. It is believed that the CCDM could provide a solid base for the development of a FLOSS cadastre application, thus contributing to the development of cost-efficient land administration systems in countries where financial resources are a constraint.

This paper is an attempt to raise awareness on the potential of FLOSS for land administration systems. The use of FLOSS for building sustainable systems makes a lot of sense. In the FLOSS world, there are no boundaries, and no divisions between rich and poor. Anyone can benefit, and everyone can contribute.

TABLE OF CONTENTS

1. INTRODUCTION

No digital land administration system can ever said to be complete. Driven by the need to keep up with user demands, security issues and increasingly complex infrastructure that requires 3D solutions, cadastre and land registration systems are pushed to develop further. Some systems are more mature than others, but even advanced cadastre systems with online enquiry services can do more to integrate better with other systems, for example to enable cross-border enquiries within the European community. At a time when European cadastres are being restructured to keep up with user demands and e-governance policies, many developing nations are building up digital cadastre systems for the first time, either by converting paper based systems to digital format or by building up a digital land register from scratch. With the growing sophistication and availability of Free / Libre and Open Source Software (FLOSS), it is the right time to re-evaluate the software needs for cadastral systems and the role that FLOSS could play.

1.1 Background

This paper was written as part of the FLOSS initiative of the FAO Land Tenure and Management Unit (NRLA) in cooperation with FIG Commission 7 and the World Bank Thematic Group on Land Administration. This paper formed an input to the FAO-FIG-WB Expert Meeting held on 22-23 October in Rome.

FAO Land Tenure and Management Unit (NRLA) assists member nations and other FAO units in the analysis, policy formulation and design of strategies to improve access to land and other natural resources and to increase tenure security for environmentally sound and sustainable rural development. Primary responsibilities are related to improving access to land and other natural resources and improving tenure security. NRLA's normative entity 2KA05 defined for MTP 2006-11[1] is called *sustainable and affordable systems, including security of tenure, for access to land and other natural resources*. It will aim on the one hand to produce materials to develop sustainable, affordable systems of access to land and other natural resources, and on the other hand to produce guidelines and methods to develop sustainable, affordable systems of land tenure security. The four main areas of focus will be the low-cost land tenure security, good governance in land administration, post emergency land institutions and land information for the poor.

FIG Commission 7 is dealing with the development of pro poor land management and land administration and has set up a working group for the application of innovative technology in land administration. One of the main tasks of the working group is to identify innovative and low cost technology to support the development for pro poor land management and administration.

The World Bank's Thematic Group on Land Administration is promoting equitable access to land and its efficient and sustainable use through policy advice, technical assistance, and lending for poverty reduction and sustainable development. The main goal of the Thematic Group is to provide a forum to exchange ideas and experiences, provide technical and financial support to studies on priority issues and new approaches in land policy.

[1] Medium Term Plan, see: http://www.fao.org/mtp/

1.2 Focus and methodology

The main focus of this paper is to examine availability and suitability FLOSS for cadastral systems and solutions, and to seek ways to support cadastral system development in settings with poor resources. First, the concepts of FLOSS in general and FLOSS for geospatial applications are discussed, as well as the functioning of open source developer communities. Then, cadastral systems and their software requirements are analysed, after which an overview is given of available open source alternatives. The rationale and potential benefits of the use of FLOSS in different economic contexts are discussed, and a number of projects and organizations are presented where FLOSS is used to some extent. The paper finishes with recommendations for FAO and partner activities in the field of FLOSS for cadastre applications.

The information in this paper is based on a review of literature, internet resources and software documentation, as well as enquiries with key informers. The author wishes to thank all those who assisted in the preparation of this paper.

2. THE ROLE OF FLOSS

The selection of software plays a central role in the development of digital cadastral systems. The choice of software has implications on the performance, flexibility and reliability of the system. If the primary aim of cadastre and land administration is to provide tenure security to land owners and investors, then cadastral organizations require secure and reliable systems based on sophisticated software.

Open source software (OSS) solutions are increasingly used for e-governance and geo-services in public works. Many governments now have the policy to consider OSS solutions alongside proprietary software products in IT procurement, and award contracts on a value for money basis. Especially in developing countries[2] with limited resources, financial considerations play a role in the choice of software for cadastral systems. Due to the absence of license costs FLOSS solutions are attractive to cadastral organizations in developing countries. For a long time, the skills needed to customize and fine-tune the software for use in cadastral systems have posed a barrier to organizations with limited technical resources. OSS tools have the reputation that they are difficult to install, run only on Unix-like operating systems and can be operated only through the command line. However, recent developments show that most OSS products becoming more user-friendly, with Windows installers and graphic user interfaces (GUI) similar to proprietary software.

So far, there is little evidence of FLOSS use for cadastral systems. A number of cadastral organizations use open source software to some extent, but the core of the information systems is generally built on proprietary software products that have a reliable reputation such as Oracle. The following sections address the concept and status of FLOSS in general and development of FLOSS for geo-informatics.

2.1 The concept of FLOSS

The general idea of open source software is that the source code of a software program should be available to anyone who wishes to study, modify or redistribute the program. Source code is a sequence of statements written in a programming language such as Java or C++, which can be read by computer programmers who are familiar with that language. To run the program on a computer, source code is compiled into binary code (zeros and ones) that can be read by machines only (see picture below). Proprietary software is usually sold only in binary format, and buyers do not have access to the source code. Open source software distributions include the binaries as well as the source code.

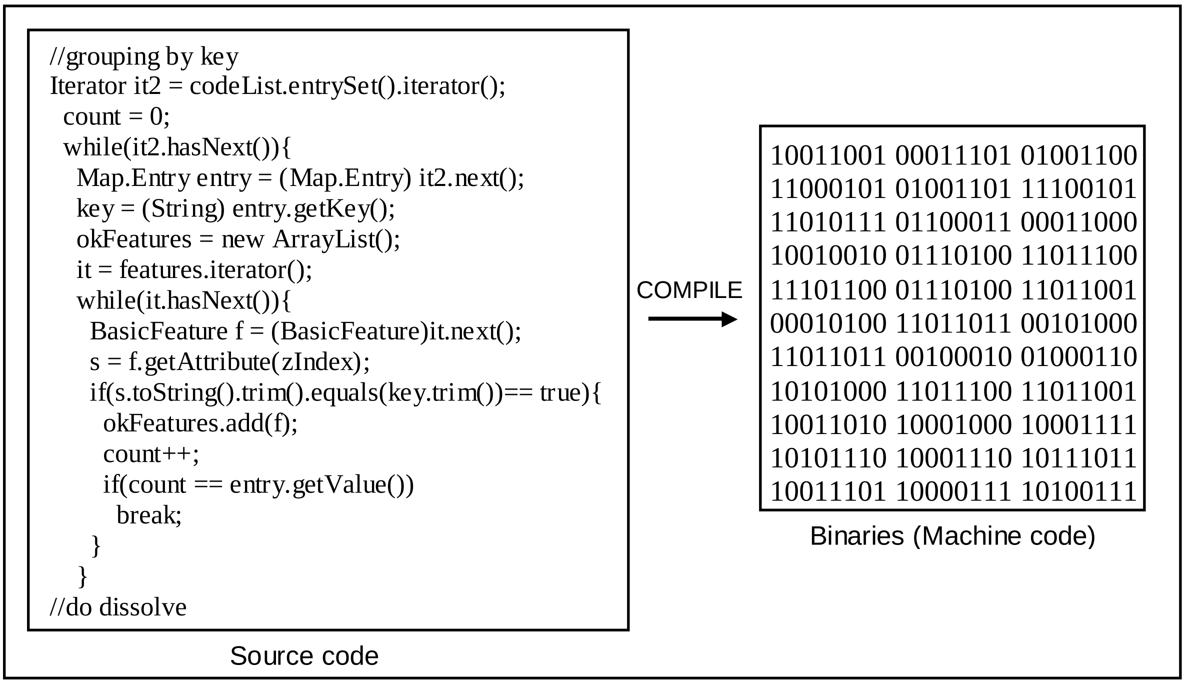
The term FLOSS was used in 2001 by Rishab Aiyer Ghosh as an acronym for Free/Libre/Open-Source Software. It is now a widely used term for the entire free software/open source phenomenon in many parts of the world. FLOSS combines the philosophies of two movements that have a lot in common: the Free Software Foundation (FSF) and the Open Source Initiative (OSI). Free software as defined by the Free Software Foundation, is software that can be used, copied, studied, modified and redistributed without restriction.[3] Without access to the source code, users would not be able to study the software

[2] Dividing the world in developed and developing countries can be considered inappropriate, seen the ongoing economic development in so called "developed" nations, and the fact that some poor countries labelled as developing countries are not developing at all. However, for lack of better terminology, the term developing countries in this study refers to countries that are less economically developed.

[3] See http://www.fsf.org/

kernel and modify it. According to the Free Software Foundation, free software is per definition open source.

The term free software should not be confused with freeware and shareware products that are widely available “for free”. In many cases, the use of these products is somehow restricted: it can be used only for a period of time, after which the product must be purchased, or it has limited functionality compared to the full-version edition which is charged for, or the software can only be used for academic purposes or personal use. Because of these restrictions, most freeware products are excluded by the FLOSS definition.[4]



Source code, binaries and the process of compiling source code into binary format

The Open Source Initiative defines open source software as software in which the source code is available for modification and redistribution by the general public.[5] Open source licenses are those licenses that comply with the Open Source Definition, the most common being GNU General Public License (GPL) and GNU Lesser General Public License (LGPL).

Research shows that the market penetration of FLOSS is high and growing. According to a study organized by MERIT in 2006, 40% of European companies report some use of open source software.[6] Also in other parts of the world the market share of FLOSS is growing. Web servers and server operating systems are the top two areas, with two-thirds of European firms using alternatives like Apache and Linux. The shares of FLOSS in other application areas are smaller, but growing. The market for desktop applications is still dominated by Microsoft with its Windows operating system and applications as MS Office and the Internet

[4] Although freeware and shareware products are not further discussed in this paper, some database and GIS freeware could be useful in land administration systems. Oracle, Microsoft and IBM have released free versions (called Express editions) of their database products. These editions have limitations compared to the full enterprise editions, but could be useful in small settings. GIS freeware products could be used for simple digitizing tasks, but since the source code is not available, they can hardly be customized to work efficiently in cadastral applications.

[5] The Open Source Definition, see http://www.opensource.org/docs/osd

[6] UNU-MERIT (2006) Economic Impact of FLOSS on innovation and competitiveness of the EU ICT sector.

Explorer, but the popularity of Open Office and Linux variations such as Ubuntu is growing. While Oracle still leads in database software, the FLOSS databases MySQL and PostgreSQL are increasingly used for Web based database applications.

Many governments have started to actively support the use of open source software in their country. In Chile, primary schools are equipped with computers running Linux operating system through an initiative of the Ministry of Education.[7] In Singapore, companies that use Linux instead of Windows are rewarded with tax credits.[8] The Brazilian government goes even further and aims to replace all Microsoft Windows installations with open source alternatives in its public agencies.[9] According to a 2006 study on government policies, 44 countries worldwide have pro-OSS policies[10] and promote the use of OSS through legislation, administrative rules or other public measures. Also IDABC[11], an institution of the European Commission, promotes the use of OSS in public sector services in Europe. According to IDABC, *OSS has several characteristics that fit particularly well the needs of public sector administrations. It allows organizations to share software and know-how and re-use it to build solutions adapted to their needs. Importantly, it can ensure adherence to open standards, thus improving interoperability and equal access to public sector information and services.*[12]

2.2 FLOSS for Geoinformatics (FOSS4G)

Seen the spatial nature of cadastre information, the interest of cadastre agencies in the FLOSS context goes especially to the development for geospatial applications. FLOSS for geoinformatics is referred to as FOSS4G, an acronym used by the Open Source Geospatial Foundation (OSGeo).[13] The history of open source GIS development starts with the development of GRASS in the 1980s. Initiated by the U.S. Army Corps of Engineers Construction Engineering Research Laboratory (CERL), GRASS became one of the first global open source software projects. Lacking financial resources to buy a commercial ESRI product that existed at that time, CERL decided to use the openness of the UNIX environment and make every effort to making it possible for anyone to contribute to the GRASS software.[14] Coordinated by CERL, developers from around the world contributed code. In 1992, the GRASS user community formed the non-profit Open GRASS Foundation (OGF). Even though GRASS was successful and started to be used by government institutions worldwide,[15] it struggled with the interoperability with other software. GRASS had an open data format, but that was not sufficient to enable data sharing between GRASS and other software products. GIS standards were needed for the sharing of heterogeneous geospatial datasets. In 1993, the Open GRASS Foundation initiated the Open Geodata Interoperability Specification (OGIS, later called OpenGIS) project which defined a vision of

[7] See http://www.iosn.net/education/news/linux-chile-schools/view

[8] Hahn, R.W. (2002) Government Policy Toward Open Source Software: An Overview. See http://www.aei.brookings.org/publications/abstract.php?pid=296

[9] See http://www.iosn.net/government/news/brazil-linux-friendly

[10] Lee, J. (2006) New Perspectives On Public Goods Production: Policy Implications of Open Source Software. See http://opensource.mit.edu/papers/LeeOSS.pdf

[11] IDABC stands for Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens.

[12] See http://ec.europa.eu/idabc/en/document/2627/5894

[13] See http://www.foss4g2007.org/about_foss4g/

[14] Westervelt, J (2004). GRASS roots. Proceedings of the FOSS/GRASS Users Conference. Bangkok, Thailand, 12-14 September 2004.

[15] In 1993, GRASS was being used by at least 25 organizations including NASA, FBI, NOAA and the Spanish, Canadian and Australian armies

diverse geoprocessing systems communicating directly over networks by means of a set of open interfaces.[16]

2.2.1 The Open Geospatial Consortium

This OpenGIS project, which is now called the Open Geospatial Consortium (OGC) forms the backbone of FOSS4G developments by setting standards for geospatial information. The OpenGIS / OGC standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT.[17] OGC works closely together with ISO and many OGC standards have become ISO standards as well.[18]

One of the most important standards for FOSS4G is the OGC Simple Feature Specification (SFS), which provides the foundations for geometry creation and editing in GIS products that comply with this SFS standard. The SFS is the base for software libraries such as the GDAL/OGR, a translator library for geospatial data formats (GDAL for raster and OGR for vector formats) used by most open source GIS products written in C++ (GRASS, QGIS, ILWIS). Also the Core Cadastral Domain Model (CCDM), which is described more in detail in section 5 (Cadastre Applications) of this study, implements the OGC Simple Feature Specification.

Other important contributions of OGC are the Geography Markup Language (GML) standard and a series of interoperability specifications for GIS on the internet including Web Map Service (WMS) Web Feature Service (WFS) and Transactional WFS (WFS-T). The OGC interoperability standards for GIS on the internet have sparked a wide range of open source initiatives. Since the development of WMS, WFS and GML, the road to internet mapping was open. GeoServer, MapServer and DeeGree are map server products focussing on internet mapping applications using WMS and WFS services. MapBuilder, MapBender, Drupal are Web GIS client products that can display map data from map servers through the WMS and WFS protocols. Also the FAO-WFP open source project InterMap[19] belongs in this category of Web GIS clients, which allow users to select map layers from several map servers, overlay them and create a customized composite map. GeoNetwork, another FAO open source project is a catalogue application with metadata editing functions for map data from web servers.[20] WMS and WFS are becoming standard functions also for proprietary GIS software and in fact, the boundaries between desktop GIS products and Web GIS clients are becoming more ambiguous.

While the evolution of internet mapping is promising, the actual editing of map data from internet web servers through WFS-T is still in an experimental stage. For now, traditional desktop GIS applications with direct database connection to SDBMS are considered the only reliable way to maintain cadastral information. The main interest of this study goes therefore to desktop GIS applications with vector editing functionality that could be used in cadastral applications. The Open Source Geospatial Foundation (OSGeo)[21] has been created to support and promote the collaborative development of open geospatial technologies and data. OSGeo

[16] See http://www.opengeospatial.org/ogc/history
[17] The Open Geospatial Consortium, see http://www.opengeospatial.org/ogc
[18] The cooperation between OGC and ISO is described in the document "Co-operative agreement between ISO/TC211 Geographic information/Geomatics and the Open GIS Consortium, Inc. (OGC)". See: http://www.isotc211.org/Agreements/Agreement_OGC.pdf
[19] See http://sourceforge.net/projects/intermap/
[20] See http://geonetwork-opensource.org/
[21] See http://www.osgeo.org/content/foundation/about.html

organizes annual FOSS4G conferences to discuss issues in the field of open source geospatial software.

2.3 FLOSS developer communities

For a long time, the open source movement has encountered scepticism. Even though the use of FLOSS is increasingly accepted, the motives of developers to dedicate their free time contributing to open source projects are questioned. A questionnaire among open source developers revealed that the main reason of developers to join an open source community is to expand and share their knowledge.[22] In his State of Open Source GIS, Paul Ramsey of Refractions Research finds that successful open source projects are created through the growth of communities of shared interest.[23]

Online communities have always been the backbone of open source development. The development of the Internet in the 1980s gave OSS development a real boost, and it is questionable whether the open source movement could have come so far without the Internet. Many OSS products have been developed due to the efforts of thousands of developers worldwide that adapted, improved and shared source code in order to develop sophisticated software products, using the internet as a medium.

Thanks to the Internet, it is not difficult to start an open source project. There are several sites that function as download repositories for open source software. Each of these sites maintains indices for querying the projects and will host information pages and discussion lists for projects that fit into their repertoire. Sourceforge.net is thought to be the largest index with more than 150.000 projects and more than 1 million registered users. Freshmeat, the second largest OSS repository, maintains an index of Unix and cross-platform software (therefore excluding software that is exclusively written for Windows operating systems), preferably released under an open source license. Savannah is a smaller repository, organized by the Free Software Foundation, and hosts projects released under (or compatible with) the GNU GPL license. The 52°North Initiative has recently started to host geospatial OSS projects and discussion boards. Apart from these repositories, many individual projects maintain their own website and discussion lists. Even though it is relatively easy to start an open source project and discussion forum, starting is of course no guarantee for success. Whether a new project will survive and maintain the interest of developers[24] will depend on the following factors.

<u>An initial product that is of interest to a group of users</u>

Some OSS projects start out of a hobby, some others as part of a research project at a university, others start out of need for certain software that is not available on the market. Some projects may start simply because it is a challenge to build it yourself instead of buying the software. But for an online developers community to evolve around a project, an initial amount of source code must be available that is of interest to a group of users. These users may then take the existing code and add improvements, thus starting a snow-ball effect by

[22] Blue Oxen (2003) An introduction to Open Source Communities http://www.blueoxen.com/research/00007/BOA-00007.pdf

[23] Ramsey, P. (2006) The State of Open Source GIS. Refractions Research Inc. Last revision May 26. 2006. See http://www.refractions.net/white_papers/oss_briefing/2006-06-OSS-Briefing.pdf

[24] There are several indicators to measure developer's interest. At Sourceforge.net, OSS projects are ranked by the number of times that the software is downloaded. A higher number of downloads indicate a greater popularity of the software with users and developers. Sourceforge.net also displays the number of developers that are contributing source code for the project. Another indicator is the frequency of software releases. Active OSS projects may release new software versions every few months. If the last release was several years ago, we can assume that the project has died out and there is no more interest to develop it further.

increasing the value and usefulness of the source code and attracting more users and contributors. Garnett (2005) argues that most successful OSS projects "*require an initial kick to get started*" and sees a role for governments: "*Getting the initial volume of working code is a place where government can have the greatest leverage, recognizing a general need for functionality, evaluating the market place, and contracting for a core piece of functionality which is amenable for extension*."[25] Many successful FLOSS projects could not have started without funding by governments or scientific organizations.

A group of core developers that can take the lead

To outsiders, it may seem that OSS developer communities manage themselves without apparent organization of operations. However in reality, there is always a group of core developers that are working on the kernel of the software and produce most of the source code. A case study on the development of the Apache Server software reveals that more than 88% of the source code was contributed by only 15 developers.[26] Designing the software kernel is the hardest part which requires close cooperation between developers. Apart from the development of the software kernel, it takes organizational skills to coordinate efforts and avoid conflicting contributions to the software. To coordinate the code contributions from developers all over the world, who have never met face-to-face, OSS projects require strong leaders who are trusted by the community to take decisions on the software development.

A good conceptual design

The organization of open source communities and the coordination of developer's contributions will be a lot easier when there is a good conceptual design for the software. Most successful OSS projects have been set up in a modular design, so that contributors can aid the project by writing special purpose modules that are quite independent from each other. Many OSS projects have multiple module maintainers that control the content and direction of their particular module.

Financial sustainability

While many OSS developers might contribute code in their free time, core developers will likely work full time on the software development and cannot be expected to work for free. Many starting projects that are dependent on research grants, government funds or donations are in the risk of dying out when the funds have finished. When financial sustainability is achieved for example by charging fees for the installation, customisation and maintenance of software, the chances of survival of the OSS project are much higher. To be successful, OSS projects require the establishment of organizational structures dedicated to their production.

The outcome of open source projects is often unpredictable, but when these four factors are there, chances are high that a project will succeed. Many projects have been abandoned; others are thriving on large communities. Software development is never finished. When there are no new releases, this usually means that the software project has reached a dead end, not completion. When successful, OSS development can result in a high level of sophistication.

[25] Garnett (2005) User-Friendly Desktop Internet GIS. See http://udig.refractions.net/docs/VSpaperuDig.pdf

[26] A Case Study of Open Source Software Development: The Apache Server. See: http://conway.isri.cmu.edu/~jdh/collaboratory/research_papers/apachefinal3.pdf

3. CADASTRE, LAND REGISTRATION AND DATA REPOSITORIES

Although the organization of cadastre and land registration operations will vary from one country to another, cadastral and land registry offices usually handle administrative and technical tasks to document and maintain information on land property. The FIG statement on Cadastre[27] defines cadastre as follows:

A Cadastre is normally a parcel based, and up-to-date land information system containing a record of interests in land (e.g. rights, restrictions and responsibilities). It usually includes a geometric description of land parcels linked to other records describing the nature of the interests, the ownership or control of those interests, and often the value of the parcel and its improvements. It may be established for fiscal purposes (e.g. valuation and equitable taxation), legal purposes (conveyancing), or to assist in the management of land and land use (e.g. for planning and other administrative purposes), and enables sustainable development and environmental protection.

Even though there is a strong relationship between cadastre and land registration functions, they differ in content. While the land register holds the records on right on land through deeds or titles, the cadastre contains information about land properties and their boundaries within a certain administrative area. Land registration and cadastre functions complement each other and should ideally be handled within the same system. The second statement of the Cadastre 2014 model[28] foresees an abolishment of the separation between cadastral maps and land registers. Yet in many cases, they are functioning independently in separate organizations and not always co-operating in the most efficient way.[29]

The design of digital cadastral systems must take the organization and required distribution of information into account. While new technologies allow data to be stored centrally, the cadastre and land registration functions might be implemented at local level with little cooperation between administrative areas within the same country. Or the land register might be maintained at central level, while cadastral offices maintain the graphic information locally. Many countries have an incomplete coverage, i.e. only the most populated part of the country is registered while land in more remote areas is not registered at all. Some countries organize systematic registration with the objective to achieve complete coverage of cadastral registration. For other countries this is considered too expensive and land parcels might be included when ownership transfer takes place, or on demand through sporadic registration.

Whether the information is stored centrally or decentralized in lower administrative levels, the extent of cadastral coverage (or number of registered parcels), and the way in which cadastral information is accessed and updated, all these are considerations with a direct impact on the design of the cadastral system architecture and the choice of software. A digital cadastral system that is being built up from scratch in a small pilot region of a developing country will initially require simple tools and low-cost solutions that can be extended and upgraded later on. Centralized cadastres with online information services covering large

---

[27] See http://www.fig.net/commission7/reports/cadastre/statement_on_cadastre.html

[28] Kaufmann, J. and D. Steudler (1998) Cadastre 2014 – A Vision for a Future Cadastral System, XXI FIG-Congress, Brighton, July 1998.

[29] Zevenbergen, J. (2004) A Systems Approach to Land Registration and Cadastre. Nordic Journal of Surveying and Real Estate Research, Vol. 1 2004.

administrative areas need sophisticated, scalable systems. What all cadastre systems have in common is the need for a spatial data store to keep and maintain cadastral data. In different economic settings, FLOSS can play a role. The following sections review FLOSS based data repositories that can be used in land registration and cadastre systems.

3.1 Land register database and RDBMS

Traditionally, land registers have been kept in paper books and maintained by bookkeepers. In digital systems, the records are usually stored in relational database management systems (RDBMS). Database technology has been developed since the beginning of the digital age, and has now matured into a solid way to keep and maintain digital data. Relational databases, the most common of DBMS systems, store information in tables that are related to each other through common fields and are generally queried through the Structured Query Language or SQL. Storing land records in an RDBMS gives the advantages that are associated with digital registration systems: large data volume storage, fast retrieval, multiple simultaneous updates, and easy availability of information over LANs or WANs.

There are many proprietary database products, of which Microsoft SQL Server and Oracle have the biggest market share. Oracle is regarded as the most technically advanced, but also the most expensive database product.[30] Microsoft's SQL Server is less expensive, but the main disadvantage is that it runs only on Microsoft Windows operating systems. In the FLOSS world, MySQL and PostgreSQL are the most popular, leaving other FLOSS database products far behind. PostgreSQL was originally called Postgres because it resulted from Ingres, an RDBMS in development since 1977.[31] Today, PostgreSQL is developed by a large community, while MySQL is being developed under the lead of the company MySQL AB, founded in Sweden by two Swedes and a Finn.[32]

Research shows that when comparing PostgreSQL and MySQL versus proprietary database software (MS SQL Server and Oracle), the FLOSS products are not far off from the proprietary ones and can truly compete. There have been several attempts to compare the computing speed of PostgreSQL, MySQL and Oracle, but the results are different depending on the hardware used, size of the database and kind of queries tested (insert, update, or select). Apparently, the overall differences in speed are not more than 10%. All three products are quite platform independent and run on Windows, UNIX, Linux and MacOSX platforms.[33]

When comparing the software installations, it appears that MySQL and PostgreSQL are easier and faster to install than Oracle. Oracle has the highest system requirements for installation. Because the FLOSS database products use less memory and disk space, they can be installed on older and cheaper hardware systems.[34] In the table below the main features of PostgreSQL, MySQL and Oracle are compared.

---

[30] The ORACLE Enterprise license costs USD 40.000 per disk processor. With many companies using eight processors or more, the total license costs can run into hundreds of thousands of dollars. See http://oraclestore.oracle.com/

[31] See http://www.postgresql.org/about/history

[32] See http://www.mysql.com/company/

[33] Oracle does not run on BSD operating systems, PostgreSQL and MySQL do.

[34] However, this does not mean that when using FLOSS, expensive hardware is not needed. Very large databases require more memory and disk space, and storing databases on SCSI disks will provide better performance and security than the cheaper IDE/ATA disks, regardless of which database software is used.

Properties	PostgreSQL 8.2	MySQL 5	ORACLE 10g release 2
Installation requirements	Minimum 512 Mb RAM, 200 Mb free disk space	Minimum 512 Mb RAM, 3 Gb free disk space	Minimum 1024 Mb RAM, 4.1 Gb free disk space
Platforms	Windows, Unix, Linux, MacOSX, FreeBSD	Windows, Unix, Linux, MacOSX, FreeBSD	Windows, Unix, Linux, MacOSX
ACID[35]	Yes	Yes	Yes
Referential integrity	Yes	Yes	Yes
Transactions	Yes	Yes	Yes
Stored procedures	Yes	Yes	Yes
Triggers	Yes	No	Yes
Union queries	Yes	Yes	Yes
Inner join	Yes	Yes	Yes
Outer join	Yes	Yes	Yes
Auto increment columns	Yes	Yes	No
Table aliases	No	No	Yes
User defined data types	Yes	No	Yes
BLOBs[36]	Yes	Yes	Yes
Replication	Yes	Yes	Yes
Data domain	Yes	No	Yes
Unicode support	Yes	Yes	Yes
Partitioning	Yes	No (planned for 5.1)	Yes
Maximum number of columns in a table	1600	3398	1000
Maximum number of columns in key index	16	16	32
User authentication	Yes	Yes	Yes
Password encryption	Yes	Yes	Yes
Grant / revoke user privileges	Yes	Yes	Yes
Licensing costs	Free	Free for GPL projects	USD 40.000 per disk processor (Enterprise license)

Comparison of basic database functions in PostgreSQL, MySQL and Oracle

PostgreSQL includes most of the advanced database features that Oracle has (stored procedures, triggers, user-defined data types) for quite some time, while in MySQL these features were developed only recently and still need improvement. Yet, both PostgreSQL and MySQL are considered stable and reliable database products.

MySQL is unique in that it has multiple storage engines. In the early versions, MyISAM was used, this was later replaced with InnoDB. Some advanced database functions are available only for InnoDB tables and others only for MyISAM tables. This makes it hard to compare with PostgreSQL, which has only one data engine, the Postgres storage engine.[37]

The most notable difference between PostgreSQL and MySQL is the licensing policy. MySQL AB, the company that leads the development and distribution of MySQL, releases its database product under the GNU GPL license for GPL projects. However, companies that intend to distribute an application based on MySQL without the source code (and therefore not comply with the GPL) must purchase a commercial license to use the software. PostgreSQL is released under the BSD license, which allows the source code to be used in proprietary software.

[35] ACID (Atomicity, Consistency, Isolation, and Durability) is a set of properties that guarantee that database transactions are processed reliably.
[36] BLOB (Binary Large Object) data type is used to store images, music files, video footage etc. in a database
[37] See http://www.devx.com/dbzone/Article/20743/1954?pf=true

3.1.1 Scalability

When covering large administrative areas with millions of land parcels, land registration databases require scalable systems. Usually, besides the land records that reflect the current property situation, also previous ownership information is kept in history records. With populations and the number of land owners growing in many parts of the world, resulting in subdivisions of parcels in ever smaller and increasingly valuable land parcels, land registration databases must have the space to grow.

An important scalability feature is partitioning, which allows tables to be split into smaller, more manageable components that can be stored on different storage devices. Oracle has the most advanced partitioning features, while PostgreSQL supports basic table partitioning. In MySQL, partitioning is still in an experimental phase, but it will be supported in the 5.1 release.

When the data is accessed and updated from multiple locations, replication schemes become important. Replication refers to the process of creating multiple copies of a database, to be used at locations that are not always connected to each other. In systems where the data is stored at a central location, replication might be used to maintain a copy of the database on a secondary server that acts as a mirror database, to be activated when the primary server fails. More complicated replication schemes are needed in systems in which data is stored in distributed databases and updated from multiple locations. For example, if land registration databases are stored and maintained at lower administrative levels, but need to be queried from all locations, replication can be used to maintain secondary data storage at the head office. Oracle as well as PostgreSQL and MySQL support database replication, but the Oracle replication tools are more advanced compared to the FLOSS database products.

Although both MySQL and PostgreSQL can apparently handle very large databases, there is no evidence of MySQL or PostgreSQL being used for databases of more than a few hundred gigabytes. Because Oracle is known to handle databases sizes of 100 terabytes or more, organizations that deal with large data volumes and have no financial constraints are likely to opt for Oracle. For organizations with limited financial resources, both PostgreSQL and MySQL provide a good alternative.

3.1.2 Database administration tools

While all database functions can be accessed through the command prompt, many administrator tools or graphic user interfaces (GUI) are available for MySQL, PostgreSQL as well as for Oracle. These tools make it much easier to design tables and queries, add columns and create indices and relationships. Most GUI allow database administrators to schedule backups, manage data recoveries and configure synchronization and replication schemes. Some of these tools are distributed commercially by third-party companies; others are developed as open source projects. MySQL AB releases the MySQL Administrator[38] under GPL license, as well as commercially. The most popular open source GUI for PostgreSQL seems to be pgAdmin[39], which has all the needed features to manage backups and security settings for PostgreSQL databases. Also open source GUI for the management of Oracle databases exist.

[38] See http://www.mysql.com/products/tools/administrator/

[39] See http://www.pgadmin.org/

3.1.3 Security

One of the basic functions of databases is to control access to the data. A good security design is essential for any database. Digital cadastre and land registration databases must be protected from unauthorized access and loss of data. The standard security features are quite similar in Oracle, PostgreSQL and MySQL. All three include user authentication and password encryption functions, and the options to grant or revoke privileges to specific users or groups of users. With PostgreSQL and Oracle, columns or sections of tables can be encrypted, which is useful to prevent disclosure of sensitive information.

Even with the best security configurations, databases that are exposed to the internet are especially vulnerable, and software developers are working continuously to keep up with ever more sophisticated security threats. Security updates for Oracle, PostgreSQL as well as MySQL are distributed regularly to database users.

3.2 Cadastral data and SDBMS

When comparing cadastral systems from one country to another, many differences can be found. Where land values are high and technology is not a constraint, parcel boundaries may be surveyed with up to a few centimetres accuracy. Systems that cover more remote or less populated areas might suffice with general boundaries or even a point location as indication to where the parcel is located. There are examples of cadastre systems where only the street address is used as parcel location, in which case RDBMS can be used as data store.

The common trend in digital cadastre systems however, is to represent units of property (land parcels) as polygon features (sometimes called faces). The corners or monuments form the nodes and vertices, and parcel boundaries are the edges of the polygon. Adjacent parcel polygons share the same boundary, so that any changes to that boundary affect the parcels on both sides. Every parcel polygon should have a unique ID that allows for referencing with the land registration database.

To store spatial data such as parcel polygons, a spatial repository is needed. A spatial data store can be considered the core of any digital cadastral system, as its reliability and performance directly impact the functionality of the cadastre.

By default, databases are designed to store textual information, numbers and date strings, but not spatial information like polygons, lines or geographic coordinates. To store spatial objects and perform spatial functions such as calculate the area of a polygon of any shape, databases must be spatially enabled. Today, several database products have developed spatial extensions that enable the storage and manipulation of spatial objects in an RDBMS.

Cadastral organizations that cover small administrative areas (and consequently deal with a limited number of parcel records) might choose to store spatial data in a vector format such as shapefile or DXF. Also pilot projects in developing countries might want to start with such a spatial file system as provided by many desktop GIS software. However, when data volume grows and the number of clients and transactions increase, the limitations of GIS file systems will soon appear. Compared to relational databases, GIS vector file formats cannot handle large data volumes. Querying and inserting new features will get very slow when files are larger than a few thousand polygons. Also, they cannot be simultaneously updated by multiple users; as there is no record locking system.

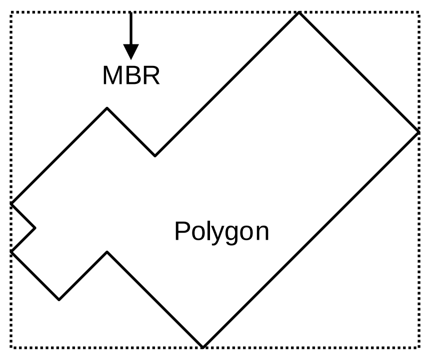
Most desktop GIS products provide database connections that allow users to store spatial data in a geodatabase or spatial DBMS (SDBMS). Without these database connections, the GIS would be unsuitable to handle a large number of cadastral records.

3.2.1 Spatial Data Engines

To store and manage spatial objects in a database, the database must be spatially enabled through a spatial data engine. ESRI's ArcSDE is such an engine that adds support for geographic objects to relational databases (Oracle, IBM DB2, MS SQL server and Informix are supported by ArcSDE). Oracle Spatial enables the handling of geographic objects in Oracle. Also IBM has developed spatial extensions for its products Informix and DB2. In the FLOSS world, spatial extensions have been developed that are considered equally adequate for the handling of spatial objects in an RDBMS. PostGIS is the spatial extension for PostgreSQL. MySQL has started to include spatial functionality in its database core since version 4.1 so that a separate spatial extension is not needed.[40] When comparing the spatial functionality of PostGIS, MySQL and Oracle Spatial, a few points are taken into account: spatial indexing, geometry and topology.

3.2.2 Spatial indexing

Spatial indexing methods influence the overall SDBMS performance and the speed at which spatial objects can be found. Accessing spatial objects is not as simple as accessing textual data in a RDBMS. Spatial indexing methods are designed to optimise the speed at which the objects are accessed. To simplify the search, spatial indexes often use the Minimum Bounding Rectangle (MBR) of geometries, which are much easier to handle than the actual geometries. MBRs that do not match the search criteria can be filtered out, after which a refined query will examine the subset of matching MBRs to determine whether the actual geometries match the search criteria.



The MBR is the rectangle that minimally bounds the shape of the feature

Traditional indexing methods were B-Tree and R-Tree. In the first versions of PostGIS, geometries could be indexed using R-Tree method, and later on the GiST method was introduced. MySQL uses R-Trees with quadratic splitting for spatial indexes on spatial columns. Oracle offers R-Tree and Quadtree indexing. Which spatial index method is best will depend on the kind of query and whether the data is being updated.

3.2.3 Geometry

The geometry model defines what kinds of geometry can be stored and how they relate to each other. The OGC Simple Features Specification (SFS) is now regarded as the standard for the development of spatial data engines. The SFS geometry types point, line, polygon, multipoint, multiline, multipolygon, and geometrycollections are supported by Oracle Spatial, PostGIS as well as MySQL. Software documentation suggests however that MySQL is not fully compliant with the OGC Simple Features Specifiation.[41] PostGIS follows the SFS for spatial data types and functions. Oracle Spatial has its own spatial model (SDO_Geometry)

[40] See http://dev.mysql.com/tech-resources/articles/4.1/gis-with-mysql.html
[41] See http://dev.mysql.com/doc/refman/5.1/en/functions-that-test-spatial-relationships-between-geometries.html

which is also compliant with OGC SFS. On the OGC website, both Oracle Spatial and PostGIS are registered with OGC as compliant with the SFS standard.[42] At the time of research for this paper, MySQL is not listed as SFS compliant.

3.2.4 Topology

Topology refers to the relationship between map features. Correct topology follows certain rules, for example that adjacent polygons cannot overlap nor have gaps between them. The last release of Oracle Spatial (10g) includes a data model and schema that store topology in the Oracle database. Also PostGIS has recently added rudimentary topology support, while full topology support is planned for the future releases. There is no indication in the software documentation of MySQL that topology support has been included.

3.3 Data repositories for land administration systems compared

This section focussed on data repositories for cadastre and land registration systems, and finds that while Oracle performs better in most aspects; both PostgreSQL and MySQL are valid alternatives for proprietary RDBMS software. When comparing the spatial functionality, it appears that as yet, PostGIS comes closer to the sophistication of Oracle Spatial than MySQL. However, MySQL is more user-friendly than PostgreSQL, which is one of the main reasons for its popularity. Both PostgreSQL and MySQL have active user communities that continuously work on the improvement of the software, and every new release is better than the previous one.

If cadastre and land registration databases could be compared, many variations in database design and system architecture would be found. Each database is uniquely designed to perform specific functions. Its table design, relationships, queries, reporting functions, record locking, data modification logs, security settings and replication functions must all be designed and configured according to the purpose of the system. But regardless whether the system is used as a legal, fiscal or multipurpose cadastre, the common need is a spatially enabled data repository with the technology to design the required functionality. The FLOSS projects PostgreSQL, as well as MySQL to a certain extent provide this technology and are acceptable alternatives for proprietary database software.

[42] See http://www.opengeospatial.org/resource/products/compliant

4. GIS TOOLS FOR CADASTRAL SYSTEMS

With the cadastral data stored in a spatial repository, cadastral organizations will need GIS tools to query, edit and maintain the data. Subdivisions and consolidations must be handled, easements registered. For data editing a large number of open source GIS tools are available, most of which are listed on OpenSourceGIS.org. Many of the GIS tools that are found here have been created for a specific purpose and have limited functionality. Some are just conversion tools, converting one file format into another. Others are meant for visualization of map files and cannot be used for editing. Several open source GIS products such as SAGA-GIS[43] and OSSIM[44] are in fact image processing products with little map editing functionality. This section focuses on desktop GIS products for the management of vector GIS data.

4.1 GIS requirements for cadastral systems

Most GIS software packages have the general GIS functions such as a variety of map symbols to display thematic layers, zooming in and out, legends, north arrows and scale bars, print functions and import / export modules in a variety of data formats. When choosing GIS software for cadastre systems, several points must be taken into account.

Editing tools

The software must have functions to create polygons, lines and point features, and to edit the geometry of polygon features by adding/deleting/moving vertices. Functions for cutting and merging of polygons are needed for parcel subdivisions and consolidations. Additional editing options such as clipping and buffering are useful when creating buffers along roads or around protected areas.

Topology

To create and maintain accurate cadastral maps, the mapping software must have functions to create and maintain correct topology. When creating features, the user must be able to set a snapping tolerance and snap to existing features. Adjacent polygons should share common boundaries and during cutting and merging, correct topology must be maintained.

Map projections

The software must support the geographic and projected coordinate systems required for cadastral maps. In most open source GIS products, the technology required for projections is drawn from the common class libraries PROJ4 (for GIS software written in C or C++) and GeoTools (for the Java based GIS products). Both PROJ4 and GeoTools provide a wide range of map projections.

Raster data

When orthophotos or satellite images are used for on-screen digitizing or verification of parcel boundaries, the software must support the raster format of the images to be used. Most GIS software products support common raster formats (TIFF, JPG), some support satellite image formats such as SPOT and Landsat and have image processing functions such as rectification, filtering and image classification.

[43] See http://www.saga-gis.uni-goettingen.de/html/index.php

[44] See http://www.ossim.org/

Compatibility with surveying data

The software should have options to import and process field data from GPS, total stations and handheld PDAs. Some GIS products have functions to connect to GPS devices and download data from them.

Presentation and output

The software must have the functionality to produce cadastral maps as required by the cadastral organization, including the required point and line symbology, additional map elements such as the legend, scale bar and north arrow, and support the printing or plotting on A0 format or custom paper size.

Database connection

When used for a considerable volume of map data, or if multiple users need simultaneous access to the same dataset, the software must support connections to access map layers stored in external databases. Database connections are also needed when linking attribute data to map layers, for example when querying ownership information through the digital map.

4.2 Overview of FLOSS desktop GIS

Seven FLOSS desktop GIS products have been reviewed to evaluate their usefulness in cadastral projects. Some have evolved from image processing and remote sensing products (GRASS, ILWIS), others are vector based GIS products (Quantum GIS, OpenJUMP, uDIG) while gvSIG seems to have originated from a CAD environment. There might be other useful FLOSS desktop GIS software, but these seven products are thought to have the most potential for cadastre projects.

The open source GIS desktop applications that are reviewed here have their individual strengths and weaknesses for cadastral purposes. Even though they have different functionalities, most of the reviewed products share common technology. The open source class library GDAL/OGR for import/export formats is used by GRASS, Quantum GIS as well as ILWIS. Both GRASS and Quantum GIS use the PROJ4 library for projections and the GEOS geometry engine. The projects that are being programmed in Java make use of Eclipse, GeoTools and / or the Java Topology Suite.

Most of the seven selected desktop GIS products are quite recent; all except GRASS and ILWIS have started during the last five years and are still in full development. Most have attracted considerable developer communities, and new versions are released regularly. When considering the use of one of these products, it is important to monitor future developments and releases. For example, TerraView (for which development started only two years ago) currently has limited editing functionality and can only be used for map viewing, but it is very well possible that it grows into a fully functional GIS within the next few years. In the following table, the characteristics of the reviewed open source desktop GIS products are presented.

Comparison of FLOSS Desktop GIS products

PROPERTIES	GRASS 6.2.2	Quantum GIS 0.8.1	uDIG 1.1	gvSIG 1.0	OpenJUMP 1.01	ILWIS 3.4 Open	TerraView 3.1.4
Type of product	Open source Desktop GIS with raster, image processing and vector analysis functionality	Open source Desktop GIS with vector and raster support, to browse and create map data	Open source Desktop Internet GIS	Open source Desktop GIS with CAD, vector and raster support	Open source desktop GIS for manipulating spatial features with geometry and attributes	Open source Desktop GIS / image processing	Open source Desktop GIS viewer
License	GNU/GPL	GNU/GPL	LGPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL
Size of download file	Multiple download files	54 Mb for Windows; 32.3 Mb for MacOSX; 23.3 Mb for Ubuntu	63.8 Mb for Windows; 74 Mb for Linux; 68 Mb for MacOSX	65 Mb for Windows, 66.8 Mb for Linux, 42.8 for MacOSX	6.7 Mb for binaries; 6.5 for source code	13.4 Mb	18.9 Mb for Linux 14.8 Mb for Windows
Operating system	Unix, Linux, MacOSX (runs on Windows only with CygWin tools)	Unix, Linux, MacOSX and Windows	Windows, Linux, MacOSX	Windows, Linux, MacOSX	Windows, Unix, Linux, MacOSX	Windows	Linux and Windows
Supported vector formats	GRASS vector (native format), read directly Shapefile, PostGIS; can import TIGER, DGN, MapInfo and GML2	OGR formats (Shapefile, MapInfo MIF/TAB, Spatial Data Transfer Standard catd.ddf, GML) PostGIS, GRASS)	Shapefile, PostGIS, OGR vector formats	Shapefile, dgn, dxf, dwg, PostGIS, WFS vector layers	JML (OpenJUMP GML) Shapefile, WKT Plugins for DXF, CSV, MIF, GeoConcept and PostGIS	ArcInfo (e00, .LIN, .PTS) Shapefile, AtlasGIS BNA, AutoCAD dxf, Infocam .SEQ	MapInfo MIF/MID, Shapefile, PostGIS, SPRING Tab/Geo, AtlasGIS BNA
Vector creation	Point, line, boundary, centroid (v.edit module)	Point, line, polygon	Polygon, line, point, rectangle, ellipse (Supports polygon with holes)	Point, multipoint, Line, arc, polyline, polygon, rectangle, circle, ellipse	Point, line, polygon (also polygon with holes, multiPoints, multiPolygon, multiLine)	Line segments, points. To create polygons, line segments must be created first which can then be converted to polygons	No vector creation
Editing functionality	Add/delete/move vertices; merge and break lines; copy/move/delete/ flip vector features; Dissolve polygons Trim/extend lines	Includes a GRASS toolbox for editing GRASS layers	Add/delete/move vertices	Add/delete/move vertices; move, rotate or flip features, Clip, dissolve	Inserting and delete vertices; cut polygon, merge polygons	The geometry of polygons cannot be edited; segments can be edited, split and merged	No editing possible
Buffer (around point, line or polygon)	Yes (v.buffer)	Only with PostGIS layers	No	Yes	Yes	No	Yes
Vector Overlays (Union, Intersect, Subtraction)	Yes	No	No	Yes	Yes	No	Yes
Spatial queries on vector layers	Yes (v.distance)	No	No	Yes (nearest neighbour / contained in)	Yes (intersects, contain, assign data by location)	No	Yes (assign data by location)
Convex hull	Yes	No	No	Yes	Yes	No	No

PROPERTIES	GRASS 6.2.2	Quantum GIS 0.8.1	uDIG 1.1	gvSIG 1.0	OpenJUMP 1.01	ILWIS 3.4 Open	TerraView 3.1.4
Snapping and topology tools	Snap function to snap one line to another while digitising. Module v.build to build topology.	The software allows the user to set the snapping tolerance and snap to nodes and vertices	Snap radius can be set to snap to nodes and vertices. With the validation plugin, vector layers can be checked for correct geometry, self-intersecting or overlapping lines and dangling nodes.	The software allows the user to set the snapping tolerance and snap to nodes and vertices. Snapping to elements from different layers is also possible.	Vertices can be snapped to the reference grid, as well as to other vertices or lines. Topology validation tool to check for valid OCG geometry	Line segments can be checked for self overlap, dead ends and intersections.	No
Supports digitising tablet	Yes	No	No	No	No	Yes	No
Raster support	More than 40 supported raster formats (through GDAL)	TIFF, ERDAS (.IMG) ArcInfo ASCII Grid SDTS (.ddf) DTED Elevation raster (.dt0) USGS DEM AIG GRASS	TIFF, JPG, GIF	TIFF, JPG, ECW, MRSID	TIFF, GIF, JPG, ECW and PNG	ArcInfo (ASCII, e00), GIF, ERDAS (.GIS / .LAN) IDA (.IMG), IDRISI (.IMG)	TIFF, JPG, SPRING Grid Files, Binary Raw Files (.RAW) ESRI ASCII Grid files (.TXT)
GPS tools	Import waypoints, routes, and tracks from a GPS receiver or GPS download file into a vector map (v.in.gpsbabel)	Supports GPX format, download from/ upload to GPS through GPSBabel	Plugin for GPS formats is planned	No	No	Supports GPS GARtrip (.txt) format	No
Database / SDBMS support	DBF, ODBC, MySQL, PostGIS/PostgreSQL SQLite	PostGIS/PostgreSQL	PostGIS, OracleSpatial, ArcSDE and MySQL	PostGIS, MySQL	Plugins for PostGIS, ArcSDE	ODBC supported databases	MS Access, MySQL, PostGIS, Oracle
Programming language	ANSI C	C++	Java	Java	Java	C++	C++
Development platform / class libraries	Software components depend on multiple libraries	Qt with PROJ4, GEOS, SQLite, GDAL/OGR and other libraries	Eclipse RCP with GeoTools libraries	Eclipse RCP with GeoTools and JTS libraries	Java Topology Suite (JTS)	MFC Library (GDAL/OGR Library for import / export module)	TerraLib, Qt
Command line / Menu bar	Both	Menu bar	Menu bar	Both	Menu bar	Both	Menu bar
Interface language	Translated into 20 languages, more coming	Translated into Czech, Slovak, Spanish, and Latvian. Translations in German, Italian, Romanian, French, Polish underway.	English, German, Spanish, French	Valencian, Spanish, Gallego, English, Czech, German, Basque, French, Italian, Portuguese and Chinese	English, Finnish, Portuguese, French, German, Spanish	English	Portuguese, English
GML	Yes	Yes	Yes	Yes	Yes	No	No
WMS	Yes (r.in.wms)	Yes	Yes	Yes	Yes	No	No
WFS	Yes (v.in.wfs)	Yes (WFS plugin)	Yes	Yes	Yes	No	No
WFS-T	No	No	Yes	No	No	No	No

4.2.1 GRASS

The Geographic Resources Analysis Support System (GRASS) is probably the best known and oldest FLOSS GIS. The development of GRASS was started in the 1980s by the US Army Corps of Engineers.[45] The complexity and UNIX dependency makes it hard for Windows users to become familiar with GRASS. The GRASS developers are working hard to make it run on Windows (promised for version 6.3) but so far, GRASS is very much a UNIX program. GRASS was originally written as a raster image processing system, and vector analysis capabilities were added later. Compared to other FLOSS desktop GIS, GRASS has by far the most sophisticated GIS and map editing capabilities.

4.2.2 Quantum GIS

Quantum GIS (or QGIS for short) is under development since 2002 and provides a more user friendly alternative to GRASS. Thanks to the close cooperation between GRASS and Quantum GIS, GRASS functionality is available from within the QGIS application.[46] GRASS layers can be edited directly without the need for conversion, although the editing options in QGIS are quite limited compared to GRASS. Although most of the common raster formats are supported, QGIS does not have the image processing functions that GRASS has. Even though QGIS is very usable and promising, it is still a very young project, which has not yet reached its version 1 release.

4.2.3 uDIG

Development of the User-friendly Desktop Internet GIS (uDIG) was started in 2004 by Canadian-based Refractions Research, the same company that is taking the lead in the development of PostGIS.[47] In contrast with other open source desktop GIS, uDIG is released under GNU Lesser General Public License. This means that uDIG can be bundled with proprietary software and redistributed commercially, although changes that are made to the source code must be contributed back to the community. Compared to other FLOSS GIS products, uDIG lacks quite a few options that are normally expected in a GIS such as buffering or calculating the area of a polygon. On the positive side, uDIG allows for direct editing of data from transactional Web Feature Services (WFS-T) and is therefore a powerful internet client tool.

4.2.4 gvSIG

This is an open source GIS product developed in Spain by IVER Technologías in cooperation with the Valencian government and the Jaume I University of Castellón since 2003. The name gvSIG[48] is a Spanish abbreviation that stands for *Generalidad Valencia Sistema de Información Geográfica*. GvSIG has close ties with CAD software, which becomes apparent when reviewing its editing functionality. There is for example the scaling option that allows the user to enlarge or reduce the size of objects, a function that is not normally used in real-world mapping, but common in CAD systems. GvSIG has an interesting feature called the commands stack, which is a tool that allows the user to undo/redo several commands at once. This is especially useful when digitizing complex features. Once accustomed to the interface, gvSIG is a user-friendly GIS product for which many translations are available. The combination of editing capabilities, platform independency and support for PostGIS as well

[45] See http://grass.itc.it/devel/grasshist.html
[46] See http://www.qgis.org/
[47] See http://udig.refractions.net/
[48] See http://www.gvsig.gva.es/index/

as MySQL database connections would make gvSIG a highly useful GIS base for FLOSS land administration systems.

4.2.5 OpenJUMP

JUMP (JAVA Unified Mapping Platform) was initially developed in 2002 by Vivid Solutions Inc. on initiative of the British Columbia Ministry of Sustainable Resource Management to do automated matching ("conflation") of roads and rivers from different digital maps. It evolved into a platform independent GIS with strong editing capabilities. JUMP uses the Java Topology Suite (JTS) which implements the OGC Simple Features Specification for geometric operations. When Vivid Solutions stopped the further development of JUMP, the project was continued by the JUMP user community as OpenJUMP, with help of a few former Vivid Solutions employees.[49] Apart from OpenJUMP, a few other projects have originated from the JUMP software. SkyJUMP, PirolJUMP and DeeJUMP each have their own specific functions. A Spanish version of JUMP is being developed under the name KOSMO.

4.2.6 ILWIS

The Integrated Land and Water Information System (ILWIS) is desktop GIS & Remote Sensing software, developed in the Netherlands by ITC up to its last release (version 3.3) in 2005. As per July 1st, 2007, ILWIS software is available as open source software under the 52°North initiative (GPL license). Its powerful image processing functions make it a highly useful tool for natural resources management and for organizations that need to process orthophotos or satellite imagery for base mapping. However, ILWIS seems less suitable for digitizing and manipulate parcel boundaries. The boundaries must be digitized as line segments, after which they can be converted to polygons. To add labels, a point layer must be created to which labels for polygons can be attached. The geometry of polygons cannot be edited, and ILWIS could therefore not handle parcel subdivisions as required in cadastral applications.

4.2.7 TerraView

TerraView is being developed by INPE (the Brazilian National Space Institute) since 2005 and released as open source GIS viewer under the GPL license. The primary objective of TerraView is to demonstrate the capabilities of the open source TerraLib class library which is also developed by INPE[50]. Although TerraView does not have vector creation functionality, vector layers can be manipulated and overlayed, creating new output layers with the results. Interesting features are the geocoding functionality to locate street addresses and the spatial statistics module. INPE also distributes SPRING as closed source freeware GIS which has more editing functionality than TerraView.[51]

4.3 Desktop GIS for cadastre systems; FLOSS and proprietary products compared

When comparing the FLOSS desktop GIS products with proprietary GIS software such as ESRI’s ArcView version of ArcGIS, the most notable differences are related to the editing functionality. None of the reviewed FLOSS products can match the number of editing options that ArcView has. Even GRASS, the most mature of the FLOSS GIS systems does not have all the editing options that are provided in ArcView, probably due to its origins as an image

[49] See http://openjump.org/
[50] See TerraView http://www.dpi.inpe.br/terraview/index.php and TerraLib http://www.terralib.org/
[51] Sistema de Processamento de Informações Georeferenciadas, see www.dpi.inpe.br/spring/english/index.html

processing system rather than a vector GIS. Options such as cutting (subdividing) and merging of selected polygons that are standard in ArcView and indispensable for the registration of land consolidations and subdivisions are only provided in JUMP. On the other hand, several of the FLOSS products provide the topology validation tools that ArcView lacks. Even though topology tools are included in ArcEditor and ArcInfo, the more expensive versions of ESRI's ArcGIS, they seem quite rudimentary compared to the CAD-like precision of gvSIG.

The advantage of most of the reviewed FLOSS GIS products is that that they provide direct connection to PostGIS layers stored in PostgreSQL, and can therefore work as a front-end to manipulate spatial data from an open source SDBMS. Furthermore, most FLOSS GIS products can run on open source operating systems as well as on Windows and MacOSX, and give therefore much more flexibility to combine different software components in one system. The availability of interface translations for the FLOSS products, sometimes up to 20 different languages, is an obvious advantage when using the software in non-Anglophone countries.

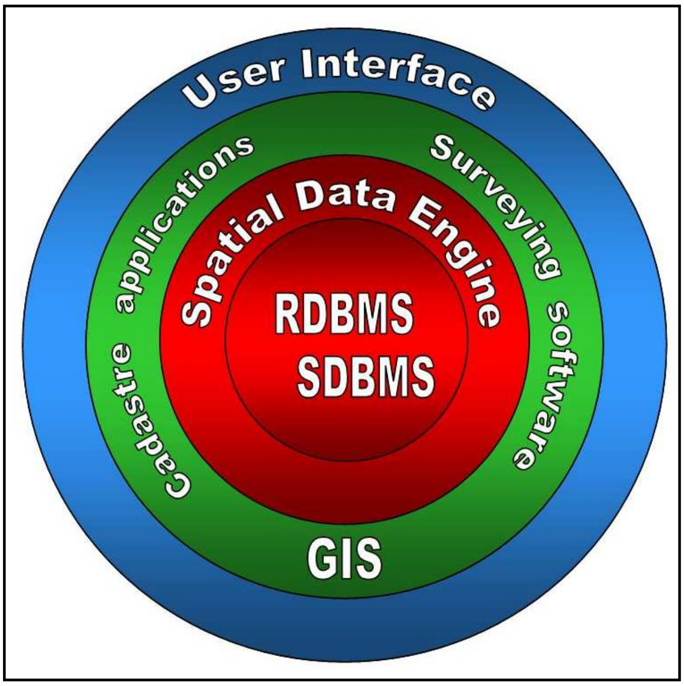
Some of the products reviewed in this section have evolved into mature GIS products and offer a real alternative to proprietary GIS software. But to use any of the reviewed open source GIS products in a digital cadastre system, additional mapping functionality would be needed to perform typical cadastral functions. The next section addresses this needed additional functionality by focussing on cadastral and surveying applications for digital cadastre systems.

5. CADASTRE AND SURVEYING APPLICATIONS

The previous chapters addressed FLOSS options for RDBMS, SDBMS and GIS software which form the core of land registration and cadastre systems. Yet more than database and GIS software is needed to build a digital system for cadastre and land registration. This section focuses on cadastre and surveying applications that, together with GIS software, form the mapping functionality for digital cadastral systems.

5.1 Software components for digital cadastre and land registration systems

Although many variations exist, the diagram below gives a theoretical representation of the software components on which cadastre and land registration systems are built. As explained in section 3 (Cadastre, land registration and data repositories), cadastral systems that cover very small jurisdictions might suffice with GIS file systems and desktop database software, without the need for a spatial data engine to store cadastral map data in a back-end RDBMS. For larger data volumes however, server-based RDBMS software with a spatial extension is needed in most cases.



Software components of cadastre and land registration systems

The data repository, which forms the core of the system, is represented in red and includes RDBMS, SDBMS and spatial data engines.[52] The user interface is the outer layer, the visible part of the system through which users interact with the software and data. In reality, digital cadastre and registration systems may have multiple user interfaces for different functions and different groups of users. For example, there might be an interface for cadastral officers to record transactions, an interface through which banks can access information on mortgages, and an online information service for public enquiries. When comparing cadastre

[52] As explained in section 3 (Cadastre, Land Registration and Data Repositories) some part of the data might be stored in RDBMS and another part in SDBMS or it could all be stored in one large spatially enabled database. There are many variations in system architecture and database design, but all digital land administration systems have similar needs for RDBMS/SDBMS database software.

and land registration systems from one country to another, the user interfaces will have little in common and reflect local implementation of land administration regulations. This outer layer, which is impossible to systematize due to the differences in land administration worldwide, is beyond the scope of this paper. The following sections address the mapping functionality, represented in green in the diagram, which includes GIS, cadastral and surveying software.

5.2 Surveying applications

The accurate recording of cadastral boundaries through surveying lies at the base of cadastral information systems. While in some cases orthophotos or other imagery might be used to define parcel boundaries, cadastral mapping in high land value zones requires surveying with GPS and/or total stations. GIS software used in cadastral systems must be able to read and process survey data and incorporate surveyed points as an integral part of cadastral data. Cadastral parcel boundaries together with survey records are often referred to as the *cadastral fabric*. In the software documentation of its Survey Analyst product, ESRI defines cadastral fabric as follows:[53]

Cadastral Fabric is a topologically integrated geodatabase dataset designed to store both a continuous parcel fabric that covers a jurisdiction as well as survey-based subdivision plans without loss of any information in the original survey record.

Surveying software may consist of extensions or plug-ins for GIS software or an independent application, and is used to transfer data from GPS and total station devices to be processed in a GIS. It may also have functions to perform survey computations such as traverse and least-squares adjustments using the original raw observations.

Several GIS vendors have started to include surveying functionality in their desktop GIS products. The problem is that there are many data formats, and it is hard for developers of GIS software to keep up with all these formats. The GPX (or GPS eXchange) format has been developed as an open standard format for the interchange of GPS data since 2002, but not yet all surveying devices can produce the GPX output format.

5.2.1 FLOSS surveying applications

Although the surveying software market is a specialized area dominated by commercial vendors, some FLOSS developments are worth mentioning. GPSBabel[54] is a FLOSS product that reads, writes and manipulates GPS waypoints in a variety of formats, including GPX and all the Magellan and Garmin formats. GPSBabel can be used on almost any operating system including Windows, Linux and MacOSX. GRASS and Quantum GIS have included the GPSBabel functionality in their latest versions so that almost any GPS format can be read directly in GRASS or Quantum GIS.

The SurveyOS[55] project defines itself as a non-profit organization dedicated to fostering cooperation between land surveyors and GIS professionals. Among the activities are the development of open source surveying tools for OpenJUMP and the development of technical surveying standards.

[53] See http://www.esri.com/software/arcgis/extensions/surveyanalyst/index.html
[54] See http://sourceforge.net/projects/gpsbabel and http://www.gpsbabel.org
[55] See http://surveyos.sourceforge.net/

5.3 Cadastral applications

Cadastral applications are closely related to surveying applications and may have overlapping functionality. Many cadastral organizations have built their own software tools for the creation, manipulation and maintenance of parcel and boundary information. GIS desktop products do not have cadastral functionality by default and need customization or software extensions to handle land consolidations and subdivisions as required by the cadastral organization. When parcel polygons are modified, the cadastral organizations will usually want to keep the original geometry archived either in a history map layer, or with a special status code in the same layer. During the modification process, the modified geometry should have a temporary status. When the consolidation or subdivision is approved, the modified geometry becomes permanent. Otherwise, the modifications are cancelled and the original geometry restored. The temporary status of modified parcels can be maintained for a long time, as long as the cadastral authorities need to approve the transaction. Multiple transaction processes can be ongoing simultaneously.

No open source applications that were specifically built for cadastre are currently found on the internet. On the search engine of sourceforge.net[56] (which lists over 150,000 OSS projects including 169 GIS and 604 mapping software projects), the keywords cadastre, land registration or LIS result in zero hits.

Commercial vendors have released software applications for cadastre that work as extensions on GIS software. Examples are the Parcel Manager for Intergraph Geomedia; ArcCadastre of ESRI, and the Parcel Editor developed by NovaLIS technologies for ESRI's ArcMap. Also Bentley has recently developed Bentley Cadastre as a cadastral extension for Bentley Map to work with Oracle Spatial databases. These cadastral applications generally support the subdivision and consolidation of land parcels, the registration of easements, handling of survey data from field measurements, and support for database queries. ArcCadastre also includes workflows for the registration of sales transactions, subdivisions and other cadastral functionalities. The cadastral application usually comes with a standard data model, which, according to the vendors, is completely customisable and can be further developed into higher levels of detail. It would be interesting to compare the data models from the different vendors, to see how well these applications fit into different national contexts and what kind of customisation is needed for individual cadastre systems.

The development and sales of proprietary cadastral applications proves that there is a demand for this kind of software, and that the needs for cadastre functionality are apparently quite uniform throughout the world. Developers of these applications have followed existing cadastral models for the design of the software functionality, or they might have created their own models. While the development of these commercial cadastral applications has been ongoing, a new comprehensive cadastral model has been designed, which might be a more solid base for cadastral software suitable for implementation in cadastral systems across the world.

5.4 The Core Cadastral Domain Model

The Core Cadastral Domain Model (CCDM) has modelled the cadastral domain using Unified Modelling Language (UML). The conditions during the design of the model were that it

- should cover the common aspects of cadastral registrations all over the world,

[56] See http://sourceforge.net/index.php

- should be based on the conceptual framework of Cadastre 2014,
- should follow the international ISO and OGC standards, and
- should be as simple as possible in order to be useful in practice.[57]

The CCDM has been developed since 2002 by members of the Dutch Cadastre, TU Delft and ITC Enschede based on experiences in existing cadastres in the Netherlands, El Salvador, Bolivia, Denmark, Sweden, Portugal, Greece, Australia, Nepal, Egypt, Iceland, and several African and Middle Eastern countries. The model is based on accepted standards for geometry and topology published by ISO and OGC, and has been submitted for ISO certification. It would be unrealistic to expect that the CCDM will fit every cadastral system in the world, but adopting the CCDM standard would give the advantages of interoperability and international cooperation in the development and functioning of cadastre systems across the world.

5.5 An open source cadastre application

Based on the CCDM, an open source cadastre application could be developed by adding cadastral functionality to an existing open source desktop GIS. The CCDM could be used for the design of a standard cadastral fabric, in the same way as the OGC SFS is used as a geometry standard for many open source GIS products.

More research would be needed to decide which product could provide the best GIS base to build a cadastral application on. Even though GRASS is the most complete FLOSS GIS, it is also quite complex and not as user-friendly as other FLOSS GIS products. GvSIG is thought to be a good candidate with its advanced editing capabilities and connections to both PostGIS and MySQL. As a Java-based and OGC compliant product, gvSIG provides interoperability, platform independence and internet mapping functionality. Also OpenJUMP could be a possible GIS base, since it already has functions to cut and merge polygons, provides connections to PostgreSQL and PostGIS, and has been chosen as a GIS base for the development of surveying functions through the open source SurveyOS project.

Without additional cadastral and surveying functions, the GIS products that were reviewed in section 4 do not have sufficient mapping functionality to fully function in a digital cadastre system. Combining open source GIS with proprietary cadastral software is not an option, since each of the available commercial applications will only work in combination with GIS software of the same vendor.

The development of additional cadastral and surveying functionality is seen as a condition for FLOSS GIS to be useful in cadastral systems. Although individual cadastre and land registration projects might well be able to develop this needed additional functionality for open source GIS products, it could be beneficial to join efforts and develop cadastral extension software for FLOSS GIS, based on the CCDM and OGC SFS standards. In this way, FLOSS GIS and cadastre software could become a better alternative to existing proprietary software for cadastre systems. After all, the FLOSS world is a world of contributing and sharing, a world of give and take.

---

[57] Oosterom, P. van, C. Lemmen, T. Ingvarsson, P. Van der Molen, H. Ploeger, W. Quak, J. Stoter and J. Zevenbergen (2006) The Core Cadastral Domain Model. Computers, Environment and Urban Systems 30, p. 627-660.

6. RATIONALE AND POTENTIAL BENEFITS OF USING FLOSS FOR CADASTRE AND LAND REGISTRATION

When considering the use of FLOSS for cadastre and land registration systems, many factors play a role. This section discusses the situations where the use of FLOSS would be beneficial, and which considerations must be taken into account.

6.1 FLOSS for land administration systems in developing countries

In places where financial resources are a major constraint in the design of digital land administration systems, the use of FLOSS seems justified. And indeed, instead of equipping all municipality offices with expensive GIS licenses to manipulate cadastral data, it is worth considering FLOSS products that may be equally adequate. The software can be downloaded freely and installed immediately, and forms therefore a lower barrier compared to procurement procedures for proprietary software. Yet the use of FLOSS for cadastral systems in developing countries is often ruled out because of the lack of technical resources. For most software users, the access to source code is not an obvious advantage, because even when they have access to it, they would not know how to modify it, let alone compile it for normal use. Customizing open source software certainly requires programming skills, which are not always available in situations where technical as well as financial resources are scarce.

However, developing digital cadastre and land registration systems in developing countries is never an easy task, regardless whether FLOSS or proprietary software is used. Each land administration system is unique, and careful designing of the database and software interface is needed to make the system work according to the laws and regulations of the local cadastral authorities. Programming skills are always needed to make a cadastral system work as it should, and choosing proprietary, closed source software could make it harder to fit the local requirements. Unless it comes with a good development platform (such as ArcObjects for ArcGIS) closed source software might be hard to customize for cadastral applications. Using open source software brings the obvious advantage that it *can* be modified to meet local requirements.

The use of FLOSS for land administration systems in developing countries would be most successful in a context where governments actively support the use and development of open source software. By promoting the use of FLOSS in education, research and public institutions, governments can provide a framework for IT development that benefits all sectors of the economy. In such a context, local IT companies are stimulated to develop innovative solutions that fit local needs. Involving the local IT sector in the development of open source software solutions for land administration systems can contribute to the sustainability of these systems.

6.2 FLOSS for land administration systems in economically advanced countries

Most cadastral organizations in richer economies have already built their digital cadastre system, using mainly proprietary software components. License fees are usually not the main concern, as the costs of licenses only represent a small part of the total cost of development and maintenance of the system. In existing systems, switching to an open source environment would likely bring additional costs that may exceed the costs of license fees for proprietary software products. When considering an open source migration, the costs related to for example the redesigning software for Linux platform that previously ran on Windows, and training of personnel in the use of Open Office and other open source applications, cannot be underestimated and should be included in the cost-benefit analysis. For many sophisticated systems, switching from proprietary software to OSS-based solutions will be relatively expensive and unpractical.

However, the use of FLOSS might be considered when digital land administration systems are upgraded or re-engineered. As all digital information systems, also digital land administration systems go through a life-cycle of five to ten years, after which they must be re-engineered to keep up with customer demands and new security requirements. At this point, the software needs might be re-evaluated and the use of FLOSS could be found favourable when considered on a value for money basis. An example of this situation can be found in Germany, where land registry offices of two federal states have opted for Linux as platform for their systems (see section 7.1).

The use of FLOSS for land administration systems can be expected to increase especially in the area of online geo-information services. Internet mapping has undergone enormous technological advances and spatial information is now abundantly available to everyone with an internet connection.[58] As citizens have come to expect information from government agencies to be available online, also cadastral organizations are providing information over the internet to customers. Especially when it comes to building online enquiry systems, cadastral organizations can benefit from the use of FLOSS. Open source internet map server and client applications are increasingly popular and sophisticated tools for distributing map data such as MapServer and GeoServer are freely available. Open standards such as OGC GML and the interoperability protocols WMS and WFS have opened the way for sharing heterogeneous GIS data over the internet. Simple web browsers such as Internet Explorer or Firefox can be used to view map data from WMS servers, eliminating the need for GIS desktop software to access this map data. This means that also geographic data, including cadastral information, can be distributed to the wide public over the internet.

While these tools are already widely used for online enquiry systems, the actual editing of geo-information over the internet through transactional WFS (WFS-T) is still in an experimental phase. When implemented, WFS-T would allow notaries to sketch new parcel boundaries resulting from property transactions on a digital map in their preferred GIS client software and send this new boundary information in the GML data format over the internet to the cadastral database on the WFS-T server. A 2006 study by T.U. Delft and the Dutch Kadaster explores how cadastral data from different data sources and different data models can be accessed and updated using WFS, GML and GeoServer.[59] The study reveals that while it is possible to do simple editing over WFS-T, issues such as area locking and maintaining correct topology while updating must still be solved before it can be used in cadastre applications. However, developments continue and the WFS-T protocol could radically change the way in which cadastral datasets are maintained and updated within the next few years.

---

[58] An interesting example is OpenStreetMap (OSM) on http://www.openstreetmap.org/, a free editable map of the whole world built with open source software. The project follows a wiki approach; users can contribute by adding geographic data. Other examples are the Geography Network http://www.geographynetwork.com/, Google Maps on http://maps.google.com/, Google Earth http://earth.google.com/, and Microsoft Visual Earth http://www.microsoft.com/virtualearth/

[59] Brentjens, T., M. De Vries, W. Quak, C. Vijlbrief, P. van Oosterom (2006) Updating geo-information in a heterogeneous networked environment – Experiences and evaluation of OpenGIS Web Feature Services. See: http://www.gisdevelopment.net/technology/gis/techgis_005.htm

6.3 Languages and localization

Mainstream GIS software exists mostly in English language. Although software vendors might release versions of their software in major other languages, many users are faced with the problem that there are not enough speakers of their language to make a commercial software release viable. Instead of waiting for a propriety vendor to release a version in their language, local FLOSS user groups are taking matters in their own hand and translate open source software interfaces for use in their country. GRASS and gvSIG are examples of how active communities may take part in the translation of the software interface. Especially in cadastre and land registration systems it is important that the software interfaces reflect the local language and culture of land registration.

6.4 Customer support and maintenance

An often reported reason for not choosing FLOSS products is the lack of customer support and maintenance services. Proprietary software vendors usually provide support and maintenance for their software, as a service included with the purchase of the product, or through a separate maintenance contract. With FLOSS products, support and maintenance are not so straightforward. In case of unexpected problems or data loss, there is no company to hold legally responsible, and users must rely on the OSS community to solve software issues. However, research shows that the response speed from OSS communities is actually higher than from customer support services of commercial vendors. A Dutch case study found that *"it needed less time for bugfixes from the open source community than from regular software suppliers. Moreover, the community makes the problem more transparent, which means a better notion of the time/effort needed for possible solutions can be obtained."*[60] Increasingly, support and maintenance for FLOSS is formally organized through companies that charge for their services. Linux, PostgreSQL and MySQL have technical support services in place that match or surpass those of proprietary software vendors. This allows FLOSS users to benefit from the convenience of customer services while profiting from lower overall costs due to the absence of license fees.

6.5 Security of FLOSS compared to proprietary software

An important concern when considering FLOSS for cadastre and land registration systems is whether FLOSS can provide the security that is required for these systems. Digital land administration systems, especially those that require the updating of cadastral records over Wide Area Networks (WAN) and include online information services, should be just as secure as online banking systems. Security of open source software is an often debated subject among developers. It has been argued that a system based on closed source software components is more secure, because when the source code is hidden from the attacker, it will be harder to find vulnerabilities in the system. In reality however, hiding the source code does not protect a system from hackers or other malicious threats.

To attack a system, hackers tend to use dynamic techniques, by sending data and see if the program response indicates a common vulnerability.[61] For example, in a logon screen for a banking service, a hacker might try to send a command statement such as GetAccountInfo(353889897) instead of a user account name, and if he is lucky the program might skip password requirements and display bank account information that matches the command statement. Access to source code is not needed for this kind of dynamic hacking

[60] IDABC (2005) Dutch use open source for Geoservices in public works. See: http://europa.eu.int/idabc/en/document/3934/470

[61] See: http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/open-source-security.html

techniques. If a hacker wishes to examine the source code for vulnerabilities, he can do so regardless whether the system is based on open source software components or not. Software binaries (machine code) can be converted back to source code using disassemblers or de-compilers. The resulting source code is not useful for developers to modify the software, but it enables hackers to search for patterns and find vulnerabilities. Also machine code can be examined for patterns that indicate vulnerabilities, using special tools that circulate among hackers.

There are arguments that support the view that open source software might actually be more secure than proprietary, closed source software. When a large community has access to the source code, security flaws will likely be discovered and fixed more quickly than would be the case with proprietary software. Or, quoting the famous Linus Law: *Given enough eyeballs, all bugs are shallow.*[62] Proprietary software vendors sustain the view that the continuous reviews of developer communities give users of OSS a false sense of security. While many developers have access to the code, there is no guarantee that vulnerabilities are actually discovered and fixed. Proprietary software vendors such as Microsoft have security experts that are working full time to test the software and fix any flaws, while most OSS developers might not even bother to report or fix security flaws that they find in the code. Whitfield Diffie, a leading expert on internet security, argues that "*there is probably some truth in the notion that giving programmers access to a piece of software does not guarantee that they will study it carefully. But there is a group of programmers who can be expected to care deeply: Those who either use the software personally or work for an enterprise that depends on it. If anyone has both the right and the need to study the code and be assured of its correct functioning, it is users.*"[63] When open source software is used in systems that require a high level of security, the developers of these systems will make sure that any security flaws the software are fixed. Contributing these fixes back to the community is the only way to make sure that future releases of the software will not contain the same flaws. In an article published in 2000 by Security Focus, Elias Levy concludes that "*open source software certainly does have the potential to be more secure than its closed source counterpart. But make no mistake; simply being open source is no guarantee of security*."[64]
As all security experts recognize, there is no 100 % security guarantee for any digital information system. System administrators must work continuously on the improvement of system security to keep up with new threats and increasingly sophisticated malware.[65] Regardless whether FLOSS or proprietary software components are used, administrators of digital cadastre and land registration systems must ensure that security measures and disaster recovery schemes are in place.

6.6 Overall benefits of using FLOSS

The most cited benefits of using FLOSS are lower license fees and less dependency on proprietary software vendors. While this is true, the main benefit as observed by Ghosh and Schmidt lies in the fact that *the process of learning and adapting software enables users to become creators of knowledge rather than mere passive consumers of proprietary*

---

[62] Linus's Law was named after Linus Torvalds, the initiator of Linux.
[63] Whitfield Diffie (2003) Risky business: Keeping security a secret. See http://news.zdnet.com/2100-9595_22-980938.html
[64] Elias Levy (2000) Wide Open Source: Is Open Source really more secure than closed? Security Focus 2000-04-17. See http://www.securityfocus.com/news/19
[65] Malware refers to malicious software designed to infiltrate or damage a computer system without the owner's informed consent (Wikipedia).

technologies.[66] Câmara and Fonseca state that OSS can help developing countries master the technology of software development and support the development of applications that leverage local knowledge.[67] OSS can be used to build products that will give a large portion of the population access to information that it would not have otherwise. However, OSS development in developing countries requires strong policies and the financial support of their governments to be successful.

[66] Ghosh, Rishab Aiyer and Philipp Schmidt (2006) Open Source and Open Standards: A New Frontier for Economic Development? UNU-MERIT. See http://www.merit.unu.edu/publications/pb/unu_pb_2006_01.pdf
[67] Câmara, G. and F. Fonseca (2005) Structural Constraints in Open Source Software Development and their Public Policy Implications. Editora MundoGEO, Curitiba, Brazil. See http://www.mundogeo.com.br/revistas-interna.php?id_noticia=5295&lang_id=3

7. EXAMPLES OF FLOSS IN CADASTRE AND LAND REGISTRATION SYSTEMS

There are a number of organizations and projects that use FLOSS to some extent for land administration purposes. Below, cases of FLOSS use in cadastre and land registration systems in Germany, Brazil, Iceland, Spain and United States are presented.

7.1 Use of FLOSS in cadastral offices of two federal states in Germany

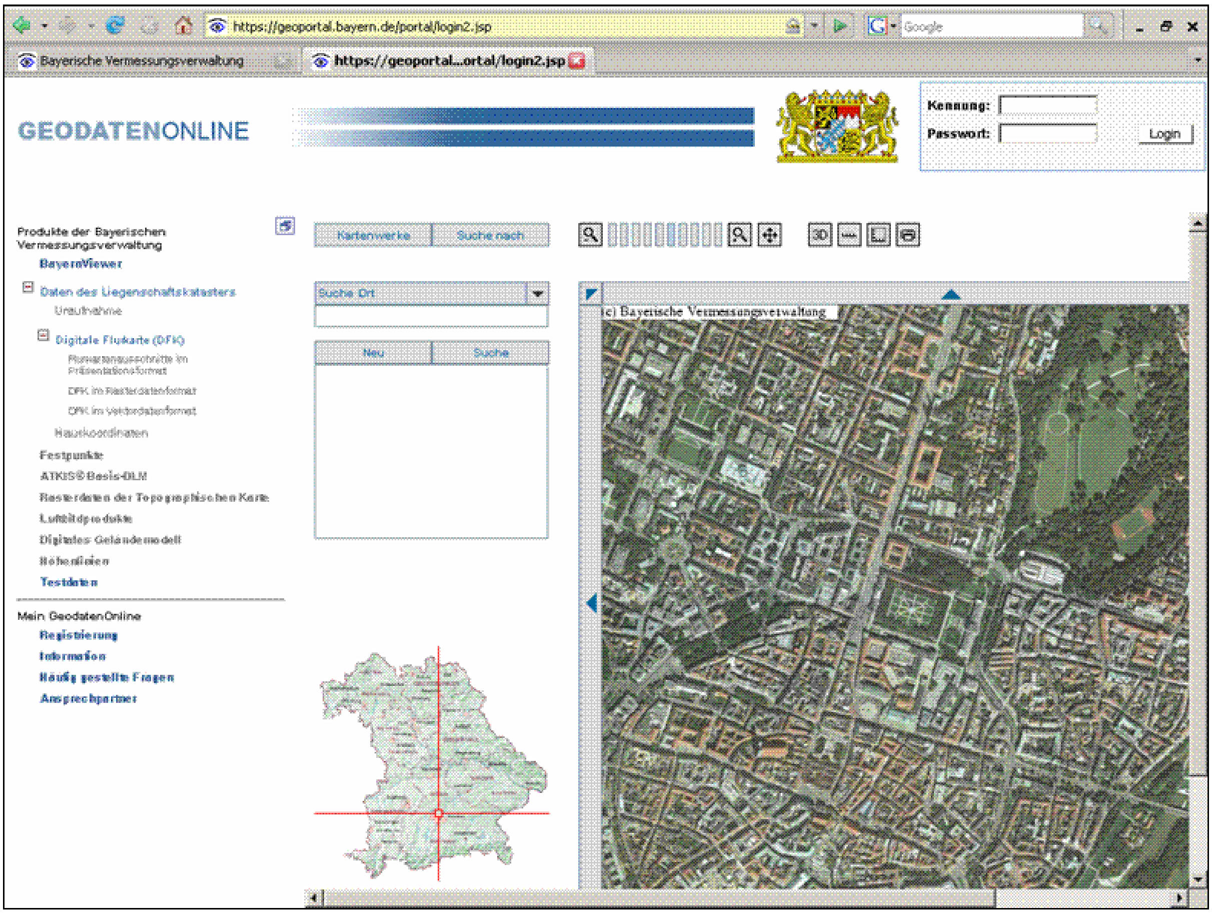
In Germany, cadastral systems are defined by the AFIS-ALKIS-ATKIS (AAA) standard which is used by all federal states. The AAA standard guarantees that the same data model is used for cadastral systems throughout Germany. However, the actual implementation and software platforms are different for each federal state. Two out of sixteen federal states (Bundesländer) of Germany, Bavaria and Rhineland-Palatinate are running their land registry systems on Linux platforms.

Bavaria is the largest German federal state with 71 districts and more than 12 million habitants. Its capital is Munich. The Bavarian Office for Surveying and Geographic Information (LVG) in Munich deals with the cadastre and land administration in Bavaria. In 2003, LVG decided to migrate from Unix to SuSE Linux platform with KDE desktop environment in its head office as well as the decentralized cadastral offices at district level. Both financial and security aspects played a role in the selection of the Linux operating system. Since Linux is based on Unix technology, the transition was relatively uncomplicated and the cadastral applications needed only minor modification to work on the Linux platform. LVG does not use any off the shelf GIS software; all applications were developed in-house based on open source software libraries. Apart from Linux and KDE, LVG uses open source web browsers (Mozilla), office and email software (OpenOffice and SendMail).[68] In all decentralized cadastral offices at district level, PostgreSQL and PostGIS are used as distributed data repository. The performance of PostgreSQL is seen as satisfactory and its reliability is appreciated. The databases are replicated regularly to the central LVG office in Munich, where it is kept as secondary data storage in PostgreSQL. Apart from PostgreSQL, Oracle is used for the maintenance of a raster database. The webportal GeodatenOnline[69], where users can browse orthophotos, topographic maps and cadastral information is built on open source technology (Apache and Tomcat webserver, PostgreSQL and PostGIS, as well as the open source applications Deegree and GeoServer).

The Surveying and Land Administration office of Rhineland-Palatinate, LVermGeo followed the Bavarian open source example and decided in 2004 to migrate their desktop applications to Linux. The main reason in Rhineland-Palatinate for the open source migration was that Microsoft would discontinue the support for WindowsNT and Office 97 which was used on client PCs. The lifecycle of server and client systems would end in 2005 and LVermGeo considered the migration to either Windows XP/Server 2003 or Linux.[70] The Linux operating system was considered of similar qualities but more cost-efficient than Windows. The transition was not easy because more than 50 software applications that previously ran on

[68] Wagner, U. (2004) Linux auf 3000 Arbeitsplatz PCs der Bayrischen Vermessungsverwaltung - Resumee des ersten Jahres. See:
http://www2.bremen.de/verwaltungsreform/Kap1/pdf/Wagner_Linux_auf_3000_Arbeitsplatz_PCs_der_Bayrischen_Vermessungsverwaltung_Resumee_des_ersten_Jahres.pdf
[69] See http://www.geodaten.bayern.de/bvv_web/blva/index.html
[70] Nagler, M. (2005) Open Source Migration of the German Surveying and Land Register Administration, MERIT, University of Maastricht. See http://ec.europa.eu/idabc/en/document/4493/470

Windows NT had to be modified to work on Linux, and staff had to be trained to work with the new software. Even though the LVermGeo saved on license fees, the costs of the migration were initially high. But according to Thönnißen, LVermGeo will benefit from the open source migration on medium and long term.[71]



The online information system GeodatenOnline of Bavarian Office for Surveying and Geographic Information (LVG) is built with open source software technology

7.2 Use of FLOSS for cadastral mapping in Brazil

Land tenure in Brazil is characterized by the concentration of large rural land properties (latifundios) that are owned by relatively few landholders. To regulate the unequal land distribution, the Brazilian government issued a law in 1999, stating that all owners of land parcels of 10.000 hectares and more had to present documentation to prove their ownership of the land within six months; otherwise the titles would be cancelled. After eighteen months, many landowners had not submitted the required documentation and as a result, the land tenure situation in large parts of the country was unclear. A study by INPE describes how its software SPRING was used to analyse the situation in the state of Tocantins, and demonstrates the use of SPRING as a tool for cadastral mapping in Brazil. [72] SPRING is not an open source product, but it is well documented freeware GIS with vector editing functionality and a useful tool for cadastral mapping and agricultural reform in Brazil.

---

[71] Thönnißen, K. (2007) Linux-Migration der Vermessungs- und Katasterverwaltung Rheinland-Pfalz, see http://www.lvermgeo.rlp.de/aktuell/heise_oss_nuernberg_20070126_dokumentation.pdf

[72] Curado, R.F. and Ferreira, E. (2003) Uso do aplicativo SPRING no auxílio à gestão fundiária: o caso do Estado do Tocantins. Anais XI SBSR, Belo Horizonte, Brasil, 05-10 Abril 2003, INPE. See http://marte.dpi.inpe.br/col/ltid.inpe.br/sbsr/2002/11.12.10.38/doc/09_076.pdf

INPE plays a central role in the open source development in Brazil, and its products TerraLib and TerraView are being used for cadastre systems in several Brazilian municipalities. TerraLib is a class library for spatial functions, and forms the base for the SIGMUN system that handles cadastral applications in metropolitan areas including Santos, São Sebastião, Caraguatatuba, São Josédos Campos, São Bernardo do Campo, Cachoeiro de Itapemirim, Vitória, and in 30 cities of the state of Bahia.[73]

7.3 Use of FLOSS and CCDM for the development of a prototype Cadastre in Iceland

A study by Ingvarsson (2005) describes the development of a prototype for the Iceland Cadastre, using only open source software and the CCDM as conceptual model.[74] In this study, Ingvarsson describes the situation in Iceland which has a long tradition of land registration, but where cadastral mapping has been largely ignored. Parcels are registered with their parcel ID but without mapping their spatial extent. Some municipalities have started cadastral mapping within their jurisdiction, but there is no overall coordination between these local efforts. The situation in Iceland is unique in the sense that three quarters of the population lives within 60 km of the capital Reykjavic. The rest of the country is mountainous with glaciers, volcanoes and sand fields, and mostly uninhabited except for coastal areas, which makes cadastral mapping relatively expensive. The Land Registry of Iceland is seeking ways to accomplish spatial delimitation of land parcels throughout the country, but it is difficult to standardize the different cadastral information systems that have been created on municipality level. The study describes the potential use of the CCDM and FLOSS products PostgreSQL, PostGIS, MapServer and uDIG for cadastral registration and concludes that CCDM and open source geo-applications can be of invaluable benefit to the development of cadastral registration in Iceland.

7.4 FLOSS for updating of cadastral boundaries by notaries in Valencia, Spain

In Spain, cadastre and land registration are handled by two separate organizations. The land register is maintained by the state secretary of justice within the Ministry of Justice. In the land register, all transactions are recorded and the information can be accessed by the public. Cadastral mapping and maintenance of parcel boundaries is the responsibility of the General Cadastral Department (Dirección General del Catastro) within the ministry of finance (Ministerio de Economía y Hacienda, or MEH). Every land parcel has a reference number that refers to descriptive information in the land register. Cadastral boundaries are kept and maintained in a cadastral information system (SIGCA) while alphanumeric data is kept in an Oracle database.[75]

Cobos et.al.[76] describe the difficulty of notaries to access cadastral map information from the General Department of Cadastre. When graphic cadastral information is not available, it is very difficult for notaries to register modifications in parcel boundaries. The project Ramon Llull was initiated in 2005 to provide notaries in Valencia the necessary tools to verify the parcel geometry of land parcels that are to be subdivided or consolidated. The system created

[73] See http://edugi.uji.es/Camara/terralib_may2004.pdf

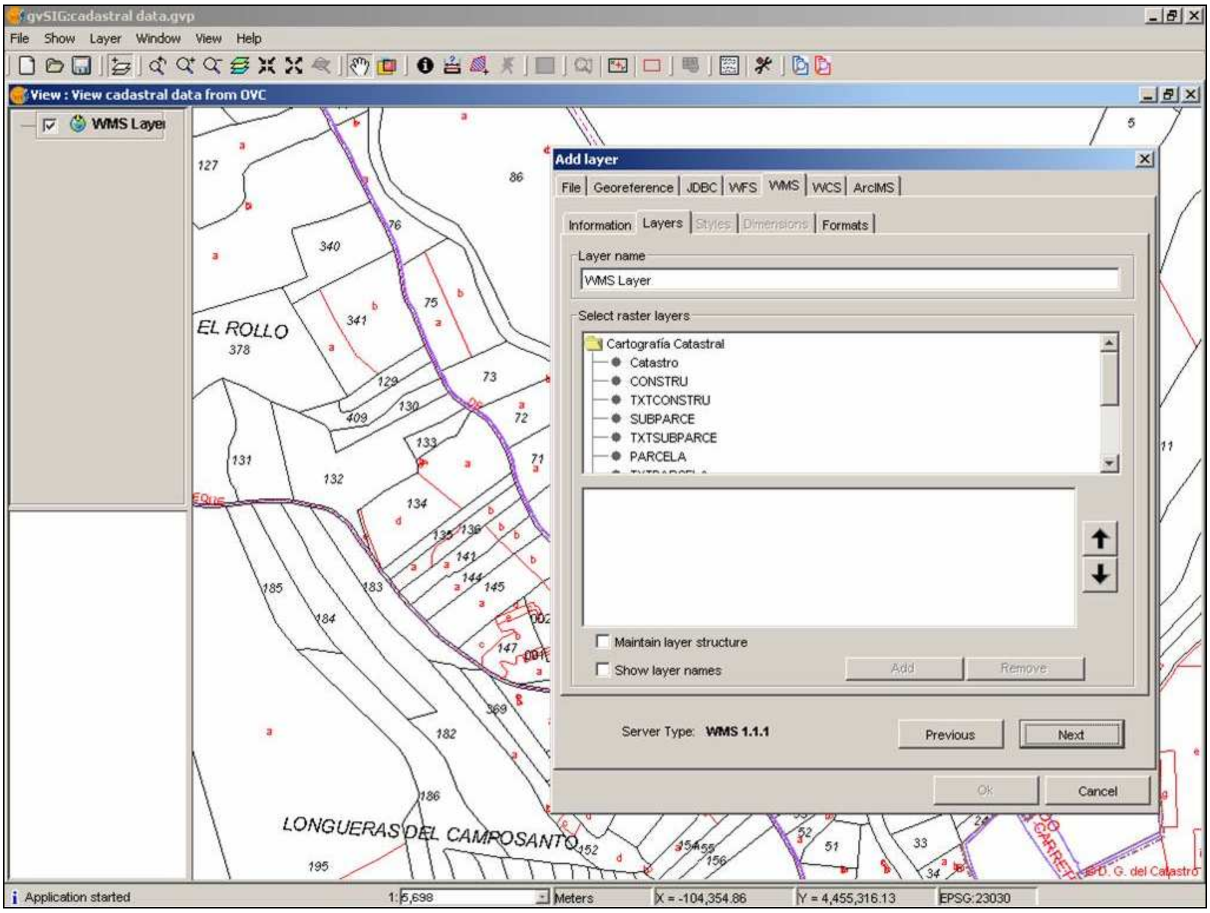
[74] Ingvarsson, T. (2005) CCDM and open source Applications in context of Implementing Cadastre in Iceland. MSc Thesis, Faculty of Engineering and Geosciences, Delft University of Technology. See: http://repository.tudelft.nl/consumption/idcplg?IdcService=GET_FILE&RevisionSelectionMethod=latestReleased&dDocName=371219

[75] Dirección General del Catastro (2000?) El Sistema de Información Catastral Español. Situación actual y estrategias de renovación informática. See http://www.catastrolatino.org/trabajos/cconejoesp.pdf

[76] Cobos, F.L., J.F. Mira Martinez, J.T. Navarro Carrión and A. Ramón Morte (2007) Proyecto Ramon Llull: sistema de gestión de alteraciones catastrales para las notarías de la Comunicadad Valenciana. Jornadas de SIG Libre, 2007. See http://www.sigte.udg.es/jornadassiglibre2007/comun/1pdf/5.pdf

by this project uses open source software (PostgreSQL, PostGIS and SharpMap) to access cadastral information stored in SIGCA from the General Department of Cadastre and enable notaries to validate the topology of modified parcel boundaries.

Accessing cadastral information in Spain has become a lot easier recently, and projects like Ramon Llull are no longer needed for notaries to obtain cadastral boundary information. In 2005, MEH opened the Virtual Cadastral Office and now, cadastral data is available as Web Map Service. GIS client software such as gvSIG (or any other GIS that can access remote data through WMS) can be used to view cadastral datasets from the Virtual Cadastral Office.[77]



Cadastral datasets from the Virtual Cadastral Office can be accessed with gvSIG, or other GIS software that supports WMS

During 2006, the Virtual Cadastral Office was visited more than 11 million times and has over 10,000 registered organizations, which are mainly notaries and land registry offices.[78] The cadastral information is updated on a daily basis, and with this Virtual Cadastral Office, MEH has a modern way of distributing cadastral information over the internet.

7.5 Online parcel information system of Ochard Park, New York

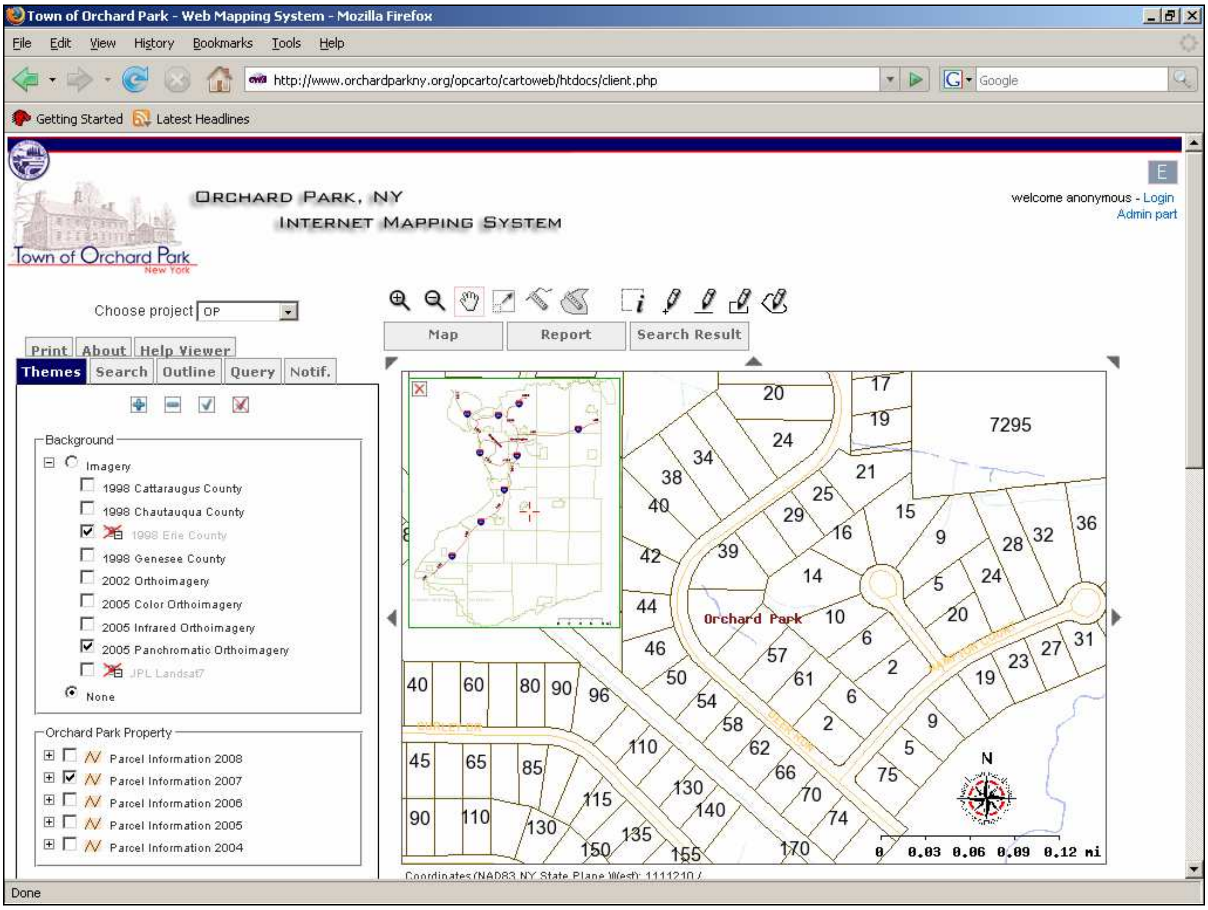
The town of Orchard Park (New York) has set up an online parcel information system with PostGIS / PostgreSQL and MapServer. On this web portal, users can access parcel

---

[77] See the website of Oficina Virtual del Catastro on http://ovc.catastro.meh.es/

[78] See http://www.catastro.meh.es/esp/publicaciones/boletin_digital/boletin_digital_32.pdf

information overlayed with Orthoimagery and topographical maps.[79] Different layers can be turned on and off, zoomed into through the zooming tools or by choosing different scales, and distances or surfaces on the map can be measured. Users can query the layers to display parcel attributes such as type of property, owner's name, parcel size, market value and sales date, and the map view with query results can be printed or saved to a PDF file.



The Orchard Park (New York) online parcel information system uses open source software PostGIS / PostgreSQL and MapServer

This web portal gives forms an interesting example of how detailed cadastral information can be distributed using open source software and OGC internet specifications, and queried through a standard web browser.

[79] Orchard Park Internet Mapping System http://www.orchardparkny.org/opcarto/cartoweb/htdocs/client.php

8. FAO AND PARTNER ACTIVITIES RELATED TO FLOSS

In the field of cadastre and land registration, FIG and FAO NRLA and the World Bank's Thematic Group on Land Administration are serving the purpose to support collaboration on the development of sustainable and affordable systems, and particularly focus on the needs of countries where technical and financial resources are limited. Especially FIG plays a role of establishing guidelines and best practices for the development of such systems. Within this context, FLOSS deserves more attention than it currently gets. Several things can be done to support the use of FLOSS in cadastre and land registration systems worldwide.

8.1 Support the use of FLOSS in land administration projects

When involved in the planning and evaluation of land administration projects, FAO and other development partners can consider the use of FLOSS products alongside proprietary software and make a choice based on cost-benefit analysis. Instead of making a cadastral institution dependent on foreign propriety software vendors, project planners may consider the involvement of local IT development efforts. When assigning international technical consultants for land administration projects, experience with the use and development of open source software technology should be among the requirements for the post. Where possible, local development of FLOSS- based solutions should be given preference over the use of imported proprietary software. The customisation and local development of FLOSS solutions may take longer to produce results, but the experiences gained from the development of such systems will benefit the local IT sector and make land administration projects more sustainable.

8.2 Support the development of a FLOSS cadastral application

While the use and development of FLOSS for cadastre systems in developing countries maybe beneficial and desirable, this is probably easier said than done. Many IT companies will have the skills for database design and the programming of interfaces for FLOSS software, but knowledge on GIS, surveying applications and cadastral functions will be much harder to find. To be useful in settings with limited technical resources, FLOSS GIS products for cadastre and land registration systems must have the basic cadastral functions that are needed for such systems. So far, none of the GIS products reviewed in this study has the needed cadastral functionality.

FIG has already made an important contribution to cadastral software development through the publication and distribution of the Cadastre 2014 and CCDM standards. These standards, if accepted, provide an important guideline for the development of digital cadastre systems across the world. Especially the CCDM model has the potential for practical implementation into a standard cadastral fabric and software functionality for cadastral systems.

By providing support to the practical implementation of the CCDM into a FLOSS cadastral application, FAO and partners could contribute to the development of sustainable pro-poor land registration systems. As described in section 4 of this paper (GIS tools for cadastral systems), several products provide useful GIS tools. Products such as GRASS, OpenJUMP and gvSIG have the basic functionality to create and manipulate GIS vector maps, with raster support for on-screen digitising. However, more development is needed to use these tools in a cadastral system. Specific cadastral functions such as the subdivision and consolidation of land parcels, the registration of easements, handling of survey data from field measurements, and specific queries for cadastral enquiries are thought to be needed in every cadastral organization and can therefore be systematized. Instead of many land administration projects developing their own cadastral applications, efforts could be bundled into a joint project, coordinated by FIG. Such a project would result in cadastral extension software for an existing open source GIS product based on the CCDM.

Financial support is needed to initiate the development of such a cadastral application. As described in section 2.3 (FLOSS developer communities), open source initiatives do not start by themselves but need an initial "kick" to produce a first working version. When a working cadastral application has been developed, the source code and binaries can be distributed, so that other developers can contribute to the improvement of the product. Even though a cadastral application can not be expected to attract a large developer community, it would be part of an already existing open source GIS community. With the availability of the source code, individual cadastral organizations could initiate the translation of the software into their own language. In this way, the product could become a better alternative to existing proprietary cadastral applications.

8.3 A forum for the exchange of experiences with FLOSS for land administration

To promote the use of FLOSS in cadastre organizations, FIG could organize an online forum for the exchange of experiences with FLOSS for cadastre applications. Through the forum, an inventory can be made of software tools that have already been created for cadastral applications and could contribute to the development of an open source cadastral application on top of an existing FLOSS GIS product. Such an online forum would not only provide a discussion platform for developers, but could also be the place where cadastral organizations in developing countries that are designing a digital system for the first time can come for support and inspiration. By raising awareness on the potential of FLOSS for cadastre and land management, FIG could contribute to the development of sustainable land administration systems.