

Presentation of a Software Application (Cadastre Toolbox) for Land Management and Administration Purposes based on Free / Libre Open Source Software (FLOSS)

Volkmar HERBST and Michael WAGNER, Germany

Key words: Free / Libre Open Source Software, FLOSS, Cadastre, Land Management, Land Administration

SUMMARY

Free / Libre Open Source Software is increasingly used in the private sector and in public administration throughout the developed countries of the world. Many software modules have been developed and for many applications the use and development of software based on the Open Source platform has proved to be economically viable. The market for GIS in countries with a low BIP is often not developed. Also the use of Open Source in these countries is mostly limited to web hosting and web applications. The prerequisite for the use of FLOSS is, however, linked with the availability of software applications which have been developed for a certain purpose. These include, amongst others, server and web applications, office applications, but also increasingly GIS applications. A cadastre toolbox based on FLOSS, which does not bind the user to a proprietary software product, could be a valuable tool to develop land administration systems in developing countries. At the same time, the user or organisation remains independent of software vendors and ITC service companies as the code can be modified by any qualified expert. This increases fair competition and will result in better implementations.

Since there is almost no FLOSS application available yet in the sector of land management and administration the company GCI (Dr. Schindler - Geo Consult International GmbH & Co. KG) has started developing a solution to fill exactly that gap. The development was initiated several months ago for a cadastre project in Senegal. The data model used is an adapted version of the Core Cadastral Domain Model of the FIG. In this way the application and database can be easily customised according to the particular requirements of any other country. Much functionality has been directly implemented in the database. This includes for instance the maintenance of a history for the data. When parcels are merged or divided, the data as it was before the change automatically becomes historical and remains in the database. Thus, the cadastral situation can be reviewed for any point in time. By implementing a lot of functionality in the database on the server the user is quite independent of particular GIS desktop software (which functions as a client to the server). The solution is mainly based on the FLOSS Database Management System PostgreSQL / PostGIS. The client components can be either Open Source desktop applications such as gvSIG, QuantumGIS, etc. or a proprietary desktop GIS like ArcView / ArcGIS.

Presentation of a Software Application (Cadastre Toolbox) for Land Management and Administration Purposes based on Free / Libre Open Source Software (FLOSS)

Volkmar HERBST and Michael WAGNER, Germany

1. OBJECTIVES

Poor land administration and land management are among the factors hindering sustainable development in many developing countries. Rapidly growing cities or the need for developing agricultural land for food production raises awareness that better land administration and land management are needed in order to make best use of the limited resource “land”.

This also involves the efficient use of spatial data for administrative and management purposes.

This issue has been taken up by the working group FIG-Commission 7 and a Cadastre Domain model was presented in the past by C.H.J. Lemmen and P.J.M. van Oosterom.

The recent development concerning standardisation in the cadastral domain is increasingly known by many organisations and private consultants who implement or support systems for land administration and land management.

The possibilities for the implementation of solutions for land administration and land management have also been changed dramatically by the availability of OGC/ISO standards and the FLOSS developments.

However there are many constraints when it comes to implementing the standards in practise. In every country the situation is different concerning the legal bodies involved, the history of cadastre and registration and the availability of land records and land boundary data.

Since 1995 the consultants of GCI have been involved in land administration and land management projects in Eastern Europe and in recent years increasingly in Asia and Africa.

Several attempts have been made to support the practical implementation of FLOSS in consulting projects and some first lessons have been learned.

One practical example, the development of the “Cadastre System Senegal”, which was developed as part of an EU funded consulting project, on behalf of Lahmeyer International, is discussed in this document.

The main aim was to develop this land administration system using the cadastre toolbox. This was the first test to see whether the toolbox could be adapted for a particular requirement.

2. CURRENT SITUATION

Currently there are few cases where FLOSS based applications are used in practice in the field of land administration. There is none to our knowledge, which is based on the Core Cadastral Domain Model (CCDM). While there are ongoing initiatives in several countries to replace or complement existing proprietary software solutions with FLOSS most of them have not been implemented yet. Thus, few examples and experiences from practical implementations are available. One such example is the Agency for Geoinformation (“Amt für Geoinformation”) of the canton Solothurn in Switzerland where FLOSS has been used very successfully¹ for several years.

An initiative towards the use of FLOSS in land administration is that initiated by FAO, WB and FIG. This led to the project “OSCAR” with the objective to develop a cadastre shell based on FLOSS². Furthermore several studies, financed by the FAO, have been carried out on the topic^{3,4}.

3. RESULTS OF THE CADASTRE PROJECT IN SENEGAL

Considering the very short timeframe of the project, the expectations concerning the developed prototype were very difficult to meet. The timeframe for the development of the documentation (analysis of processes, domain model and data model) was 3 months and the time of development of the prototype only 4 months.

Discussion of the Domain model:

The names of the classes of the domain model are modelled in French in order to be able to discuss the model with the local partners. However the following class diagrams have been translated into English for this paper.

The developed domain model was derived from the Core Cadastral Domain Model (CCDM) developed by Christiaan Lemmen, Clarissa Augustinus, Peter van Oosterom and Paul van der Molen⁵. Some changes to this model have been made in order to fit the requirements in Senegal and to increase simplicity in order to fulfil the task of developing the application within the limited timeframe. The RRR class of the CCDM (Rights, Restrictions and Responsibilities) was basically reduced to “Rights”. Some other parts of the CCDM were not implemented and some were simplified after discussions with the client.

Two different state organisations are responsible for cadastre and registration respectively. The aim was to develop a system that should cover firstly the cadastre part and secondly all

¹ Amt für Geoinformation: <http://www.so.ch/departemente/bau-und-justiz/sogis/>, checked 2008-01-30

² OSCAR: http://source.otago.ac.nz/oscar/OSCAR_Home, checked 2008-01-30

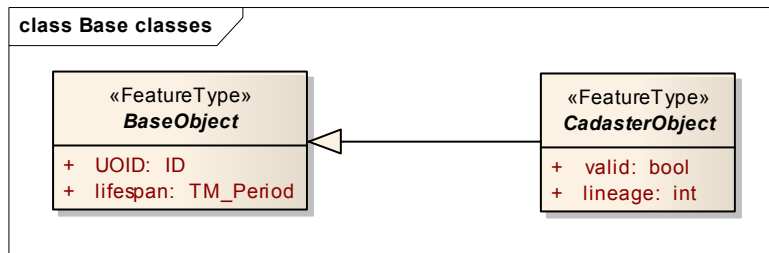
³ Pieper-Espada, Gertrude: Open for Change, Scoping Paper on the Use of FLOSS in Cadastre and Land Registration Applications. FAO, 2007

⁴ Micanovic, Wagner, Osmanovic: Feasibility Study on the Use of FLOSS in the Cadastre Sector in Bosnia and Herzegovina. FAO, 2008

⁵ Christiaan LEMMEN, Clarissa AUGUSTINUS, Peter van OOSTEROM and Paul van der MOLEN: ISO/TC 211 New Work Item Proposal, Geographic information — Land Administration Domain Model (LADM), 05.01.2008

information about rights on cadastre objects. However maintaining information about land titles (titres foncier) will not be part of the system. For this an interface should be described in order to facilitate the development of the system for the registration of land titles at a later stage.

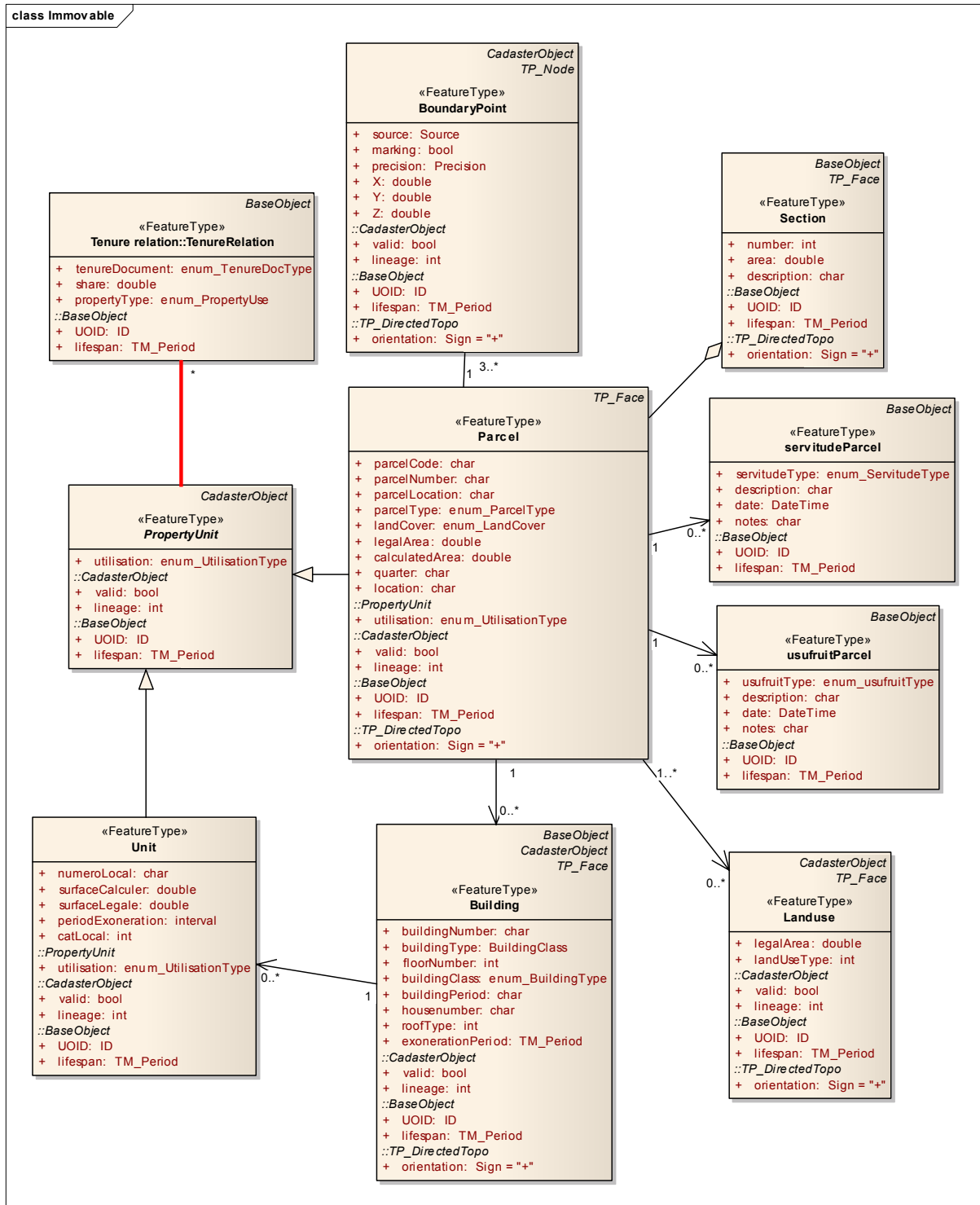
All objects in the domain model inherit from the base class. A unique ID (UOID) and a life span (TM_Period), which is used for the versioning of objects and the creation of a history, ensure that all changes to these objects can be tracked. The main cadastre objects (parcel, building and building units) inherit from the class cadastre object. Two attributes, “valid” and “lineage” are inherited by the respective classes.



Class diagram base object/ cadastre object

The attribute “valid” permits the division between valid and non-valid objects. This allows for the maintenance of preliminary information in the database, which has not been checked and “validated”. This is already important during the migration of data into the system. Existing digital data is often of poor quality, attribute information is only partly attached and has not been checked. This information can be imported into the system but will be automatically excluded from versioning and history creation. At the same time there are fewer constraints, quality restrictions and rules defined in the database for non-valid objects. Hence there are “double-standards” for valid and non-valid objects. This provides more flexibility concerning the way the cadastre agency will organise the compilation of cadastre data in the system. Systematic and sporadic data acquisition is possible. In practise it is possible to validate whole cadastre sections and import these objects as valid objects, e.g. in urban or economically active areas, or import available digitised data without all required attributes and validate sporadically e.g. during registration procedures.

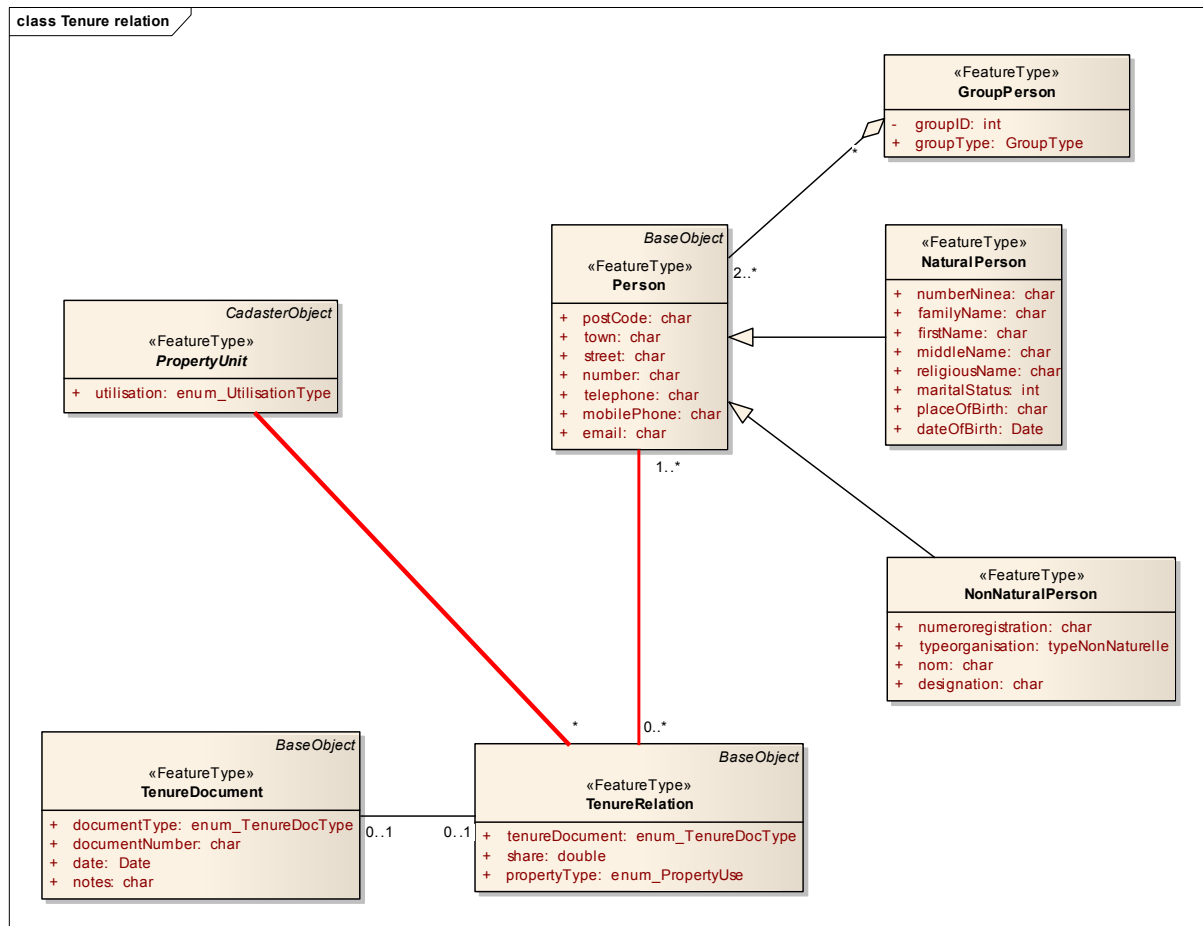
The class diagram “Immovable” describes the main cadastre objects parcel, building and building unit. Inherited attributes are displayed for any class in the class diagram. For instance the class building inherits from ::BaseObject, ::CadastreObject and ::TPFace but not from ::PropertyUnit. The inherited attributes of the class are shown in the class diagram. The objects parcel and building unit are “property objects” (inherited from class PropertyUnit) and can participate in a tenure relation with an owner, possessor or user. A building is not necessarily considered as a “property unit”. Not all buildings are registered and some are only sheds or annexes to main buildings even though there are building objects in the database. Most buildings are digitised from raster images and the status of the building is often unclear. However if the building is a registered property, a building unit will be defined which contains all information and takes part in associations.



Class diagram cadastre objects

The class “PropertyUnit” is an abstract class. Parcel and Unit inherit from “PropertyUnit” and can take part in a tenure relation. The class “TenureRelation” (class RRR in the CCDM) relates persons (in this implementation exclusively owners) to the respective property. A change in the tenure relation, e.g transfer of ownership, will trigger versioning (TenureRelation inherits from BaseClass and has a lifespan) and the history will be saved automatically.

It can also be expected that the class “TenureDocument” will be further refined in the future.



Class diagram tenure relation

Isolation of data schema and workflow schema:

There are no classes directly supporting any workflow. Even though procedures such as splitting parcels, merging parcels and transfer of ownership are common procedures which apply to all land administration systems, these can be quite different in detail. Thus it was decided to isolate the data layer from the workflow implementation. In the particular project, for instance, the procedures were not finally defined at the time of project implementation. It is also evident that the organisational structure might change, which will have some impacts on how procedures are organised and who will be responsible for certain operations. This also

includes the role of private contractors, who will either deliver data, which will be used for updates or update and validate complete datasets for the agency.

Data model:

The data model has been translated into English but its defined tables and attributes are specific to the project in Senegal and will have to be changed for land administration systems in other countries.

The model was developed for PostgreSQL/PostGis, which is used as the database for the system. In order to model versioning and n:m relationships container classes (tables) have been created (e.g. cparcel for parcel). Here is an example of where an attribute of a parcel has been changed: The land cover has been updated. Hence a new version of this parcel will be created in order to track the changes. However all related tables are untouched from this event because the foreign key (uoidparcel) will not change. The table cparcel (grouping all versions of one object) is used to relate this object parcel to the property tables (propparcel) and the owner.

Parcel table

gid (PK)	uoidparcel (FK)	parcelcode	lifetime_start	lifetime_end	landcover	...
2355	SN010000002035	0101010200100018	2006-09-16 17:53:27	2008-02-10 11:13:43	6	
2356	SN010000002035	0101010200100018	2008-02-10 11:13:43		13	

cparcel table

uoidparcel (PK)
SN010000002035

Related parcel property table propparcel

gid (PK)	uoidpropparcel	uoidparcel (FK)	uoidowner (FK)	lifetime_start	lifetime_end	part	...
1276	SN010000000339	SN010000002035	SN010000001268	2007-11-19 09:23:21		1	

Related owner table

gid (PK)	uoidowner (FK)	lifetime_start	lifetime_end	familyname	...
899	SN010000001268	2007-05-26 11:23:11		Portier	

In the same way the property table (propparcel) and the owner table can have different versions.

Database Management System:

PostgreSQL/Postgis was chosen as the database management system. Together with the PostGIS extension it proved to be well featured and flexible enough for the implementation of the requirements defined in the domain model. The aim was to maintain validation, versions and history of objects as well as other behaviour exclusively within the database. Validated objects are treated differently and in order to validate an object, certain requirements must be met.

The validation was carried out mainly through trigger functions which ensure integrity of the database during insert, update and delete procedures.

Certain objects in the database can be set as being valid. These are all objects which are inherited from the UML class objectCadastre (cadastreObject). For instance a parcel can be imported into the database but may not be a registered parcel. The attribute valid (registered) of the table or feature class will not be set or will be set to false. At a later stage if the object can be linked to an owner and a property is built and the parcel is a properly registered object the attribute valid (registered) will be set to true. Cadastre objects behave differently depending on whether they are registered or not. For instance the change of a “special” attribute will trigger a new version. Such a case is a change of the official parcel area. A new entry in the table parcel with the same parcel ID (new version) will be generated.

Behaviour	Registered object	Non registered object
unique number	SNRR0000000000 (SN010000000123)	SNRR_000000000 (SN01_0000000988)
create automatic numbers (last parcel code in section)	no	yes
create automatic section code if no section is available	no- returns an error	yes
create a new version if a special attribute changes	yes	no
create history if object expires	yes	no
prevent deletion of object	yes	no

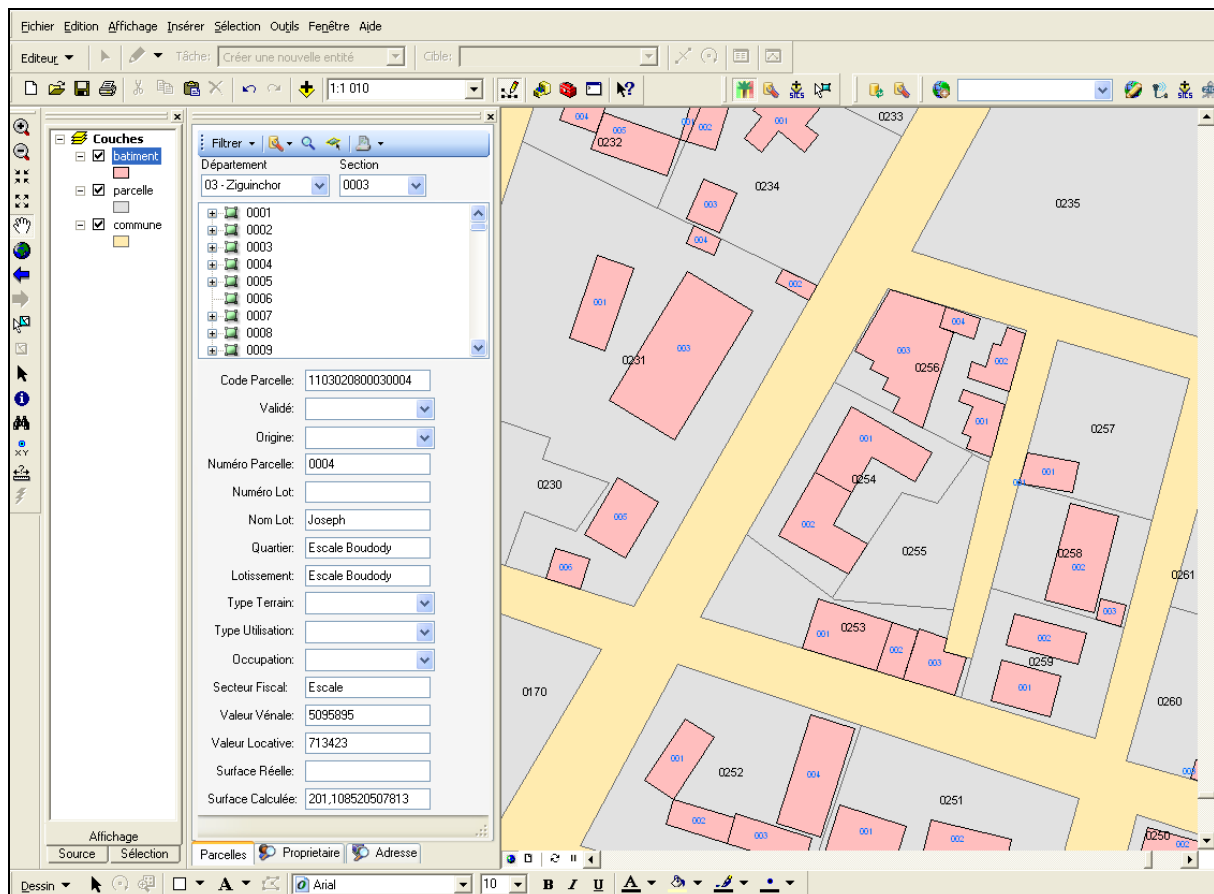
These are only some examples of the different behaviour. This behaviour can be administered in administrative tables.

A topology as defined in ISO could not be implemented throughout the database as this would require the development of a special database extension for PostGis, which was beyond the scope of this project. However the topology constraints defined in the domain model could be met by implementing verification routines in the database for individual spatial objects. For instance if a parcel has been changed or a new parcel is inserted into the database, trigger functions will automatically check for overlays of parcels and will snap boundary points of

the changed parcel to existing boundary points in a user-defined tolerance. In case of overlays of parcels the update will be rejected. Depending on the clients' needs and experiences some more topology functions and rules can be implemented.

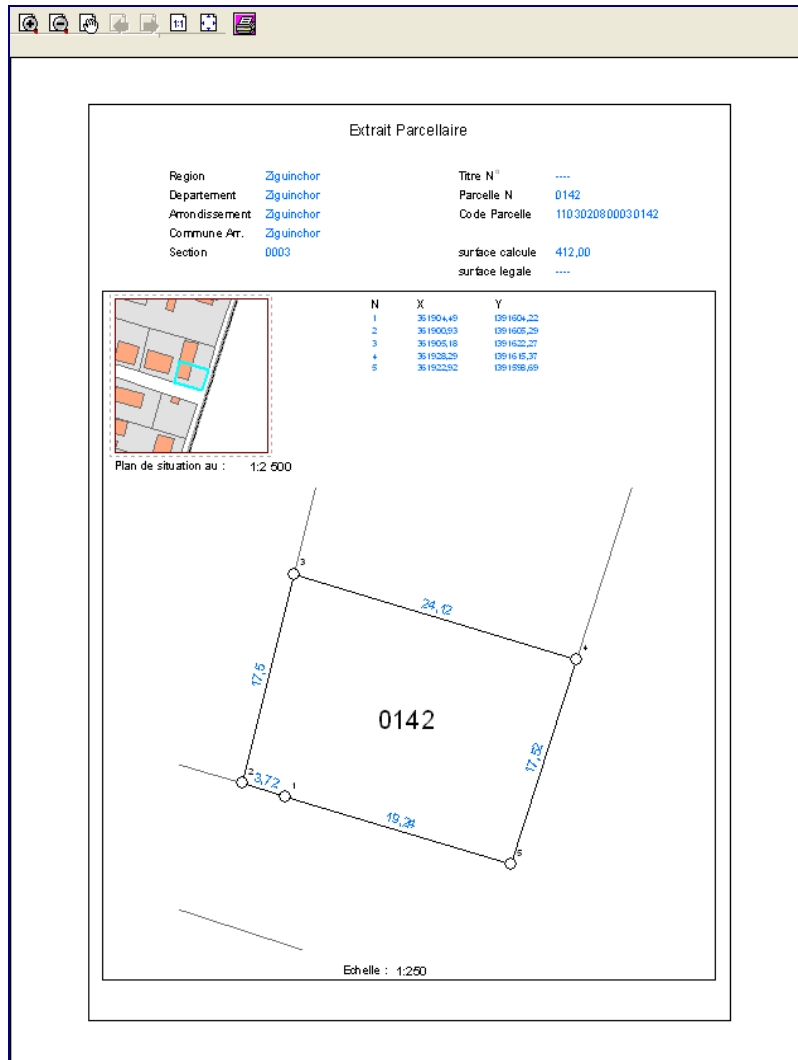
GIS client:

It was decided by the local administration not to use an Open Source client GIS application because of existing ArcGIS (ArcVIEW) licenses. Besides the existing ArcGIS/ArcView licenses training had also previously been carried out, which was an additional argument to use ArcGIS as the client application. However the development was carried out with the focus on developing additional cadastral functionality within the database through stored procedures and triggers so that at a later stage the ArcGIS client can be exchanged with an Open Source application. In order to connect the ArcGIS(ArcView) client to the PostgreSQL/Postgis database the ZigGIS connector, based on Npgsql was used. This works not only for connections of multiple clients to the PostgreSQL/Postgis database but also for multi-user editing. An ArcGIS Extension providing a user interface with additional functions was developed in C#. NHibernate, an ORM (object relational mapper) framework was used to read/write data from/to the database from the client.



Developed extension in ArcGIS

One of the advantages of ArcGIS are the layout and printing capabilities and the use of templates. Some functions were developed for automatic page layout generation and printing of parcel extracts on the basis of templates, which can be changed by the user without programming skills.



Printing of parcel extracts.

The following features/ functions have been implemented:

- Customised import of shape files and attributes:
The operator can configure the import of data in shape files into the postgis database. Field mapping and filtering of data is supported.
- Analysis and query of cadastre data:
The user interface allows for filtering parcel, buildings and building units by parcel numbers and/ or by owner names. Search functions have been implemented which also filter for certain attributes.

- Quality checks of cadastre data:
Some functions have been implemented for quality checks prior to database import.
- Output of statistics:
Statistics are calculated in the database using pgsql functions. The result is displayed in the client as database views. The user can load predefined layers in order to easily change between different representations.
- Customised printing functions:
Printing of parcel extracts and sectional maps has been implemented in order to allow the creation of parcel extracts according to the requirements of the agency.
- Customised editing workflow:
Editing of validated data is organised in cases. A case can be for instance a parcel split or parcel merge operation. Each case is a long transaction which can be rolled back in case of failure. All editing functions, which are implemented by the client, can be used for updating/ processing the geometry.

Implementation of the project and further adaptations:

In the future it is planned that information will be served from the central database to regional centres. Nevertheless the data import and quality check will be organised centrally as skilled operators are not available at regional level.

A WMS service was configured and installed in order to serve maps directly to other organisations. At the same time a GML scheme has been defined in order to allow data exchange and interoperability with other organisations, namely with the agency responsible for title registration. However these other systems are still in the conceptual or development phase and the practical tests could only be carried out on a “demonstration level”.

In the future more training for operators and for administrators is crucial.

Local consultants were involved in the development and some changes could be carried out locally. Due to the availability of the source code of all developed components any contractor could update the application. However in order to update the client component a version of ArcGIS (ArcView) is still necessary.

4. CONCLUSION

Some conclusions can be drawn about the technical feasibility of the practical implementation of the cadastre toolbox concerning the standards mentioned above and FLOSS.

One main drawback of the practical implementation of the planned FLOSS project was that the project could not be implemented as a pure FLOSS project as there were constraints defined by the recipient to be met. Thus proprietary software had to be used as a client, which possibly would also be the case in other real-life situations. However the most important aspect of the application is that the database implementation allows the connection from different clients, also FLOSS clients and the respective system can be extended using other FLOSS components such as web mapping applications.

Using PostgreSQL/PostGIS for the implementation proved to be a good choice. Only once during the development did some alternatives have to be found to a missing feature in order to simulate nested transactions.

The connection between ArcGIS and PostgreSQL/PostGIS, especially using the undocumented ZigGIS API for the development with ArcObjects was rather tricky. However all requirements could be fulfilled using the described setup.

Apart from the technical feasibility of the application there are other aspects which should be mentioned. Convincing decision makers of the advantages of FLOSS is difficult in the countries/regions where pirated software is freely available and widespread. On the other hand the absence of license costs for "Open Source Software" is a reason for underestimating the usability of FLOSS (what is for free can not be good).

One main prerequisite for a successful project is the participation of the client or recipient of the project. It is very difficult to discuss domain models on an abstract level. In order to understand the complexity of the model the notation (in our case UML) must be understood by key persons of the project partner. The development of an adapted domain model takes time and should involve different stakeholders in the land administration sector. This is often neglected or simply not possible in the given time of a development project. The local staff involved must be prepared for this task and granted enough time.

The development of a domain model and data model is a long term process which is often in contradiction to development projects which are planned as short or medium term projects. The development of land administration systems in development countries or countries in transition is no less complex then in the developed world. Often the unclear legal situation or the missing, poor or contradictory data available make the development of a land administration system very complex and difficult (often more so than in developed countries).

Local support must be involved or built up from the start of such a project and a budget secured for further development of the system after the project ends.

However if the initial development of an OSS product is pushed through and supported sufficiently (e.g. by a long term project) and an adapted and stable basic product can be provided as a result, there is a good chance that even after the end of the project and the end of professional support the software will be continuously and actively developed further by the local administration.

CONTACTS

Volkmar Herbst
GCI – Dr. Schindler Geo Consult International GmbH & Co. KG
Marienbrunnenstr. 4
04299 Leipzig
GERMANY
E-mail: herbst@gc-i.de,
Web site: www.gc-i.de

Michael Wagner
GCI – Dr. Schindler Geo Consult International GmbH & Co. KG
Marienbrunnenstr. 4
04299 Leipzig
GERMANY
E-mail: wagner@gc-i.de
Web site: www.gc-i.de