UNIVERSITY OF NAIROBI

DEVELOPMENT OF AN XML STANDARD FOR SURVEY DATA INTERCHANGE IN KENYA

PRINSTONE MWAMBURI NJUMWA

F56/33846/2019

A project report submitted to the Department of Geospatial and Space Technology in partial fulfillment of the requirements for the award of the degree of:

Master of Science in Geographic Information Systems

September, 2021

Declaration

I, Prinstone Mwamburi Njumwa, hereby declare that this project report is my original work. To the best of my knowledge, the work presented here has not been presented for a report in any other university.

Prinstone Mwamburi Njumwa .... Date 01/09/2021

This project has been submitted for examination with our approval as university supervisor(s)

....

Dr. Collins Mwange

Date 01/09/2021

Acknowledgement

The writing of this project report has been a long and arduous journey. From the initial concept to the final product, this report has seen the input from several individuals.

First and foremost, I would like to take this opportunity to thank the University of Nairobi and the department of Geospatial and Space Technology for giving me an opportunity to undertake my degree in MSc Geographic Information Systems. Great appreciation to my project supervisor Dr. Collins Mwange for offering immense support and guidance throughout the process.

Over the course of the last one and a half years I have been fortunate to meet and make friendship with different people of dynamic backgrounds. I thank all my MSc colleagues for sharing views and making the studying environment pleasant.

Finally, I am grateful to my parents for their understanding, patience and love. Without their support, I would not have been able to make it this far.

Abstract

The lack of a modern cadastral system since the establishment of surveying practice in 1903, has resulted to a lot of problems in land management and administration in Kenya. Most of the operations are carried out manually and records pertaining land are still stored in paper format. Currently, the National and County Governments have demonstrated the need to modernize the cadastre for it to enhance better land management and administration, support the establishment of an integrated NLIMS and a NSDI.

The survey plan's authoritative nature ought to be maintained in the absence of a paper document as we move towards submission of survey data in a digital form. It is for this reason that an XML standard for data encoding and interchange of a Fixed Boundary Survey Plan comes in handy to give rules and guidelines to aid in survey plan data lodging into the modern cadastral system.

The study was guided by three specific objectives: identification of the key data elements of the Fixed Boundary Survey Plan, designing of an InfraGML standard for encoding the data elements and lastly writing a web application to demonstrate the interchange of cadastral data with established GIS programs such as ArcGIS. The methodology of the study encompassed analysis of ten (10) Fixed Boundary Cadastral Plans in identifying key data elements that were later encoded in the designed InfraGML standard to provide for the demonstration of survey data interchange with established programs such as ArcGIS.

The study identified the key data elements of a Fixed Boundary Survey Plan to be the Survey Plan Number (Folio Register Number), Parcel Number, Parcel's Coordinates, Road Reserve, Easement and abuttals. An InfraGML standard for encoding the data elements was developed and demonstration of the interchange of cadastral data with established GIS programs such as ArcGIS and QGIS done using a web application. The adoption and integration of this standard with the National Land Information Management System named Ardhisasa will provide for standardization in encoding of the cadastral plan data and interchange of the same. Interoperability in storage and transferring of data between systems will also be achieved.

Table of Contents

List of Figures

List of Abbreviations

ASP – Active Server Page

B2B – Business to Business Communication

DCOM – Distributed Component Object Model

DOM – Document Object Model

DTDs – Document Type Definitions

GIS – Geographic Information System

GML – Geography Markup Language

HTML – Hyper Text Markup Language

HTTP – Hypertext Transfer Protocol

ICT – Information and Communication Technology

IFC – Industry Foundation Classes

ISO – International Organization for Standardization

IT – Information Technology

LADM – Land Administration Domain Model

LIMS – Land Information Management System

LINZ – Land Information New Zealand

MSMQ – Microsoft Message Queuing

NSDI – National Spatial Data Infrastructure

OASIS – Organization for the Advancement of Structured Information Standards

OGC – Open Geospatial Consortium

PDP – Part Development Plan

RIM – Registry Index Map

RMI – Remote Method Invocation

RRR – Rights, Restrictions and Responsibilities

SAP – Systems Applications and Products

SDI – Spatial Data Infrastructure

SGML – Standard Generalized Markup Language

SoK – Survey of Kenya

SQL – Structured Query Language

SVG – Scalable Vector Graphics

UML – Unified Modeling Language

UTM – Universal Transverse Mercator

XHTML – Extensible Hypertext Markup Language

XML – Extensible Markup Language

XSD – XML Schema Definition

XSLT – Extensible Stylesheet Language Transformation

WML – Wireless Markup Language

WWW – World Wide Web

W3C – World Wide Web Consortium

CHAPTER 1: INTRODUCTION

1.1 Background of the study

In modern societies, land management and administration is depicted by the use of technology which provides enhancement in the collection, storage, management and administration of information pertaining land. Technology also offers spatial enabling potential by the use of location as an integrating basis for information pertaining land.

Land information in Kenya is largely held in a form of paper, management is also manual and even the records in paper format are not impeccably organized. Data retrieval and dissemination in this kind of cadaster is time consuming and inefficient. This makes the system ineffective in guaranteeing timely decision making.

In most jurisdictions, the considered cadastre's basic unit is land parcel. While some countries' cadastre only encompasses land parcel, countries such as Germany captures both land parcels and buildings in their cadastre. For example, textual and graphical data description of parcels in Germany, contains all parcels and buildings in a state. According to the Sectional Properties Act, (2020) Kenyan cadastre includes buildings. Generally, the cadastre is based on parcels, i.e., information is referenced in a geographical way to unique, well-defined units of land parcels. Land parcels in state ownership like roads and streets are included in the cadastre.

Preparation of Fixed Boundary Survey Plans involves mapping land parcels to the highest positional accuracy specification (0.03m). For this reason, Fixed Boundary Survey Plans were embraced in this project in the development of the XML Framework for encoding survey data in Kenya.

1.2 Kenyan Cadastral System

In over a hundred years since the establishment of Kenyan Cadastre in 1903 to support white settlers land alienation process, the system has remained more less the same. Majority of the operations are still carried out manually and land records are stored in paper format.

Many problems facing land management and administration in the country have been as a result of lack of a modern cadastral system. However, in the recent past the Government has demonstrated the need to make the cadastre modern so as to enhance better land management and administration, support the establishment of an integrated LIMS and NSDI.

The cadastral survey plan will be a critical instrument in the development of a modern cadastral system. The fixed boundary survey plan aids in administering rights on property, since it is the beginning point in defining the location and extent of a land parcel. In addition, the plan is also a primary source of data for a geodatabase in a land information system.

The authority in essence and nature of the fixed boundary survey plans ought to be maintained in the absence of a paper document in the growing trend towards digital submission of the plans. For this reason, an XML framework for survey data interchange is critical to be set in advance in order to define properties in a right way taking into consideration of its topology, identifying them in two-dimension (2D). The standard will be based on a fixed boundary survey plan as it is a vital instrument in managing and administering rights on property because it is the cadastral map that provides a beginning point by defining the precise location and extent of the property. Fixed boundary survey plans are also a primary source of data for a geodatabase in a land management and information system.

1.3 Problem Statement

The Cadastral system in Kenya in more than hundred years since its establishment has remained largely the same. Majority of the operations are carried out manually and records pertaining to land are stored mostly in a paper format. Data is often exchanged in hard copy formats. The situation is difficult to manage both by private practitioners and quality control staff. This contributes to insufficiency of the system.

Increasingly, there is a shift towards digitization in Kenya without a proper strategy on the development of data interchange standards. For instance, in the near future, surveyors and other players using the cadastral system will be expected to digitally lodge survey data, and this may result in confusion if data interchange standards are not developed beforehand. Moreover, there are likely to be many players using the system, such as the counties and other ministries dealing with Surveying and Mapping.

1.4 Justification for the Study

Spatial data as one of the vital constituents of Land Information Systems has encountered a lot of challenges. Software developed in various departments use their own data formats hence leading to problems such as the loss of data or incompatibility on exchanging from one format to another.

Most of the studies that have been done on interoperability and invention of open data formats to aid data owners to prepare their data for all users with different needs and different software have not outlined a particular standard to guide data interchange. Strictly speaking, there is no study till date that has been done on developing a common standard for survey data interchange in Kenya. Moreover, this study contributes to development of efficient cadastral systems.

1.5 Objectives

The main objective of this study is to develop an XML standard for data encoding and interchange of a Fixed Boundary Survey Plan.

The specific objectives are:

(i) To identify the key data elements of the Fixed Boundary Survey Plan

(ii) To design an InfraGML standard for encoding the data elements

(iii) To develop a web application to demonstrate the interchange of cadastral data with established GIS programs such as ArcGIS and QGIS.

1.6 Scope of Work

The study focuses on the Fixed Boundary Survey Plan. Figure 1.4 depicts Kisii Municipality as the study area. The choice for the study area was influenced by the fact that Survey Plans for the area were easily accessible.

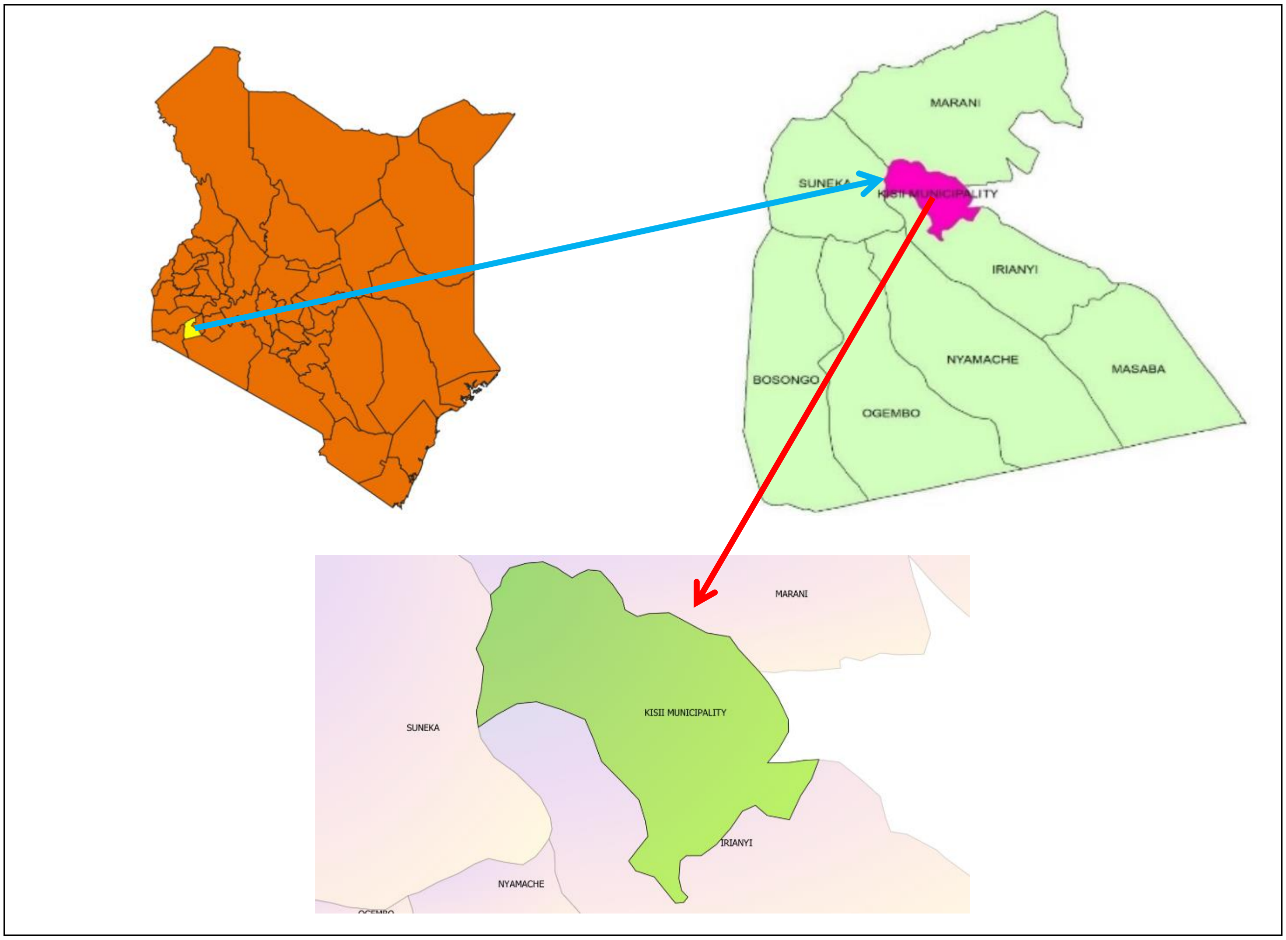


Figure 1. 1 Study Area (Courtesy, Survey of Kenya)

CHAPTER 2: LITERATURE REVIEW

2.1 Web Services

Traditional Web Applications and Web Services have some similarities but the noble function of web services is to use Hypertext Transfer Protocol in performing remote method calls as per a client need instead of just serving HTML content for display on a browser. For a long time, implementation of platforms and language specific protocols such as DCOM or Java RMI were required in order to access remote objects. These approaches however regarding the mentioned technologies cannot work well in truly distributed heterogenous systems they only work well in like for like systems i.e., in an environment that is homogeneous. The is because they used proprietary formats that could not support different kind of operating systems in connection to other technologies.

With its ubiquity. HTTP makes it suitable for transporting web services and it is perhaps the basis of the internet. Through HTTP, web services are hosted within web and application servers and are made available within a company and across the world at large. HTTP as a key application of web services therefore allows for communication between and among businesses hence the phrase Business to Business (B2B) communication.

XML as a component of web services is a well-formed text document that uses simple tags to describe data structures. XML as a text document is fully independent on either technology or platform. For example, integration of a Java program with a .NET application is made simple by XML because XML as a text document can be read by both systems and passing of any kind of data from the other program can be understood provided that they use a similar schema.

In summary, web services have two key advantages, namely;

i) Platform and language independence

The machinery or system that works to create XML behind the scenes can be using any language and not on any particular platform. The common nature of Hypertext Transfer Protocol with its ubiquity makes it the ideal and better choice for integrating systems in data interchange.

ii) Firewall friendly file transfer protocol

On any PC machine, it is through port 80 HTTP where web service traffic travels. This means that a System Administrator is relieved of any additional headaches of managing firewall like leaving a machine with extra ports open.

Therefore, in regard to these advantages, any machine that utilises technology that can read XML and has server access through HTTP, the web service and its related methods can be accessed.

2.2 Extensible Markup Language (XML)

XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. As stated by Moran, (2007) World Wide Web Consortium (W3C) committee developed XML to enhance power, flexibility and functionality than Hyper Text Markup Language (HTML) while making it easier to use and friendly than SGML.

XML is capable of getting data from any operating system, application or computer and share it with other applications, computers and internet. It gives an avenue for industries, institutions and firms to make an agreement on how and what data should be like and xml makes it plug and play.

Whether your service centers or clients are using the Feature Manipulation Engine (FME), ArcGIS or QGIS, XML provides the solution. All you need is just to hit a button “Export to XML” and an XML file is generated. The file can then be used as per the user’s wish or purpose.

Therefore, an Extensible Markup Language framework can be used to implement features to aid a software developer in developing applications to aid production of all data in XML. The framework transforms the pure XML data format document defined and produced by the programmer to any format of interest. Several data formats transformations like XHTML, Excel or Word format, SVG, WML, can be achieved by one code - one XML. Many other types of documents may also result.

Vital information from one business could be shared with several other businesses once XML is in place in a way that is more consistent. Moreover, XML data format is flexible and modifications can always be done on it to suit any circumstances in the future. This feature enables XML to consistently be up to date and remain current.

Webber, (1998) depicts Extensible Markup Language framework to consist of three major other components that include: global repositories, agents and templates.

In a nutshell, templates are rules that give guidelines on how data manipulation should be done. Templates helps software developers define the rules of how the processing of the transmitted data should be done, hence giving the developers more flexibility and relieves them of complex and long standardization processes. This is very crucial for institutions and companies that are involved in exchange and sharing of documents with each other in a very short period of time. In general, making use of Extensible Stylesheet Language Transformations (XSLT) help us define templates. They can be referenced globally through global repositories and their transmission can take place jointly with an XML message (Ives & Lu, 2000).

Agents as programs require defined instructions in the templates on how to process XML messages. XSLT file can be interpreted by agents in this case working as parsers in order to generate a HTML file established from the preceding XML message. When making use of Document Type Definitions (DTDs), agents are referred to as validators.

Global repositories can be compared to central warehouses which can provide storage for all reusable objects. These reusable objects including: DTDs, agents and templates can then be referenced or downloaded globally from the repositories (https://help.scribesoft.com, n.d.).

Moreover, XML provides the ability for developers to design and establish their own tags. With the feature in mind, programmers can develop and customize a tag if one is not available in the global repositories. If the tag is well built and regularly used it is eventually developed in the industry and stored in the repositories for future referencing (Ives & Lu, 2000).

I envision that XML coupled with DOM, XPath, and XQuery which are its associated technologies will provide a viable approach in developing an enterprise solution for Survey Data Interchange in Kenya.

As depicted in (https://help.scribesoft.com, n.d.), XML has become the de-facto standard for internet data interchange. When developing SDI’s, XML is one of the most important IT standards. The XML grammar that specifies geospatial datasets and geographical features is referred to as GML.

2.3 XML Framework Module Structure

The two parts of an XML framework module are:

- ❖ Part 1: Extensible Markup Language syntax, elements and attributes, Extensible Markup Language schema, XPath, Extensible Stylesheet Language (XSL), UML class diagrams and usage of XML parsers and editors.
- ❖ Part 2: GML, geometry simple feature elements, geometric models, Geography Markup Language schemas and spatial reference systems.

With Insight, it is possible to transfer data between systems using XML, which ties the systems or applications together using a number of different methods. Because there is no direct connection between the data source and the target, this type of integration is referred to as "loosely coupled".

To accomplish this type of integration, one should:

- ❖ Use one Integration Process (IP) to extract data from a source as XML
- ❖ Transfer the XML to the target location using a transport method
- ❖ Apply the XML data to the target application or system using a second IP.

The following available transport methods can be used to define an IP to extract data from the source system:

HTTP

With HTTP, the IP uses the XML Component to post the data to remote Server Scripts (ASP) based on either the XMLBridgeToFile.ASP or XMLBridgeToQueue.ASP sample pages. The receiving ASP page persists the XML to a file or a queue, where a File or Queue IP detects the data and applies it to the target system.

FTP

With FTP, the IP uses the XML Component to write the data to a remote FTP site. If the receiving Console has local file access to the FTP site, it uses a File IP to detect the file and apply the data to the target system. If the receiving Console does not have local access, it uses a Time IP to periodically check the FTP site for incoming files, and, when detected, applies the data to the target system.

Email

With Email the IP uses the XML Component to write the data to an email message, and then sends it to a mailbox at the target's location. When the target's location receives the email, the Email Bridge detects the new message, extracts the XML data, and writes it to an MSMQ message queue. Finally, a Queue IP detects the message and applies it to the target. Figure 2.1 illustrates the typical architecture of a web service.

As stated by Czarnik, (2013) the goal of an XML Framework is to aid:

(i) Interchange of data between systems, applications or programs with different internal data formats.

(ii) Usage of an XML interface: components integration, client/server communication nodes of distributed systems, remote configuration and monitoring of application.

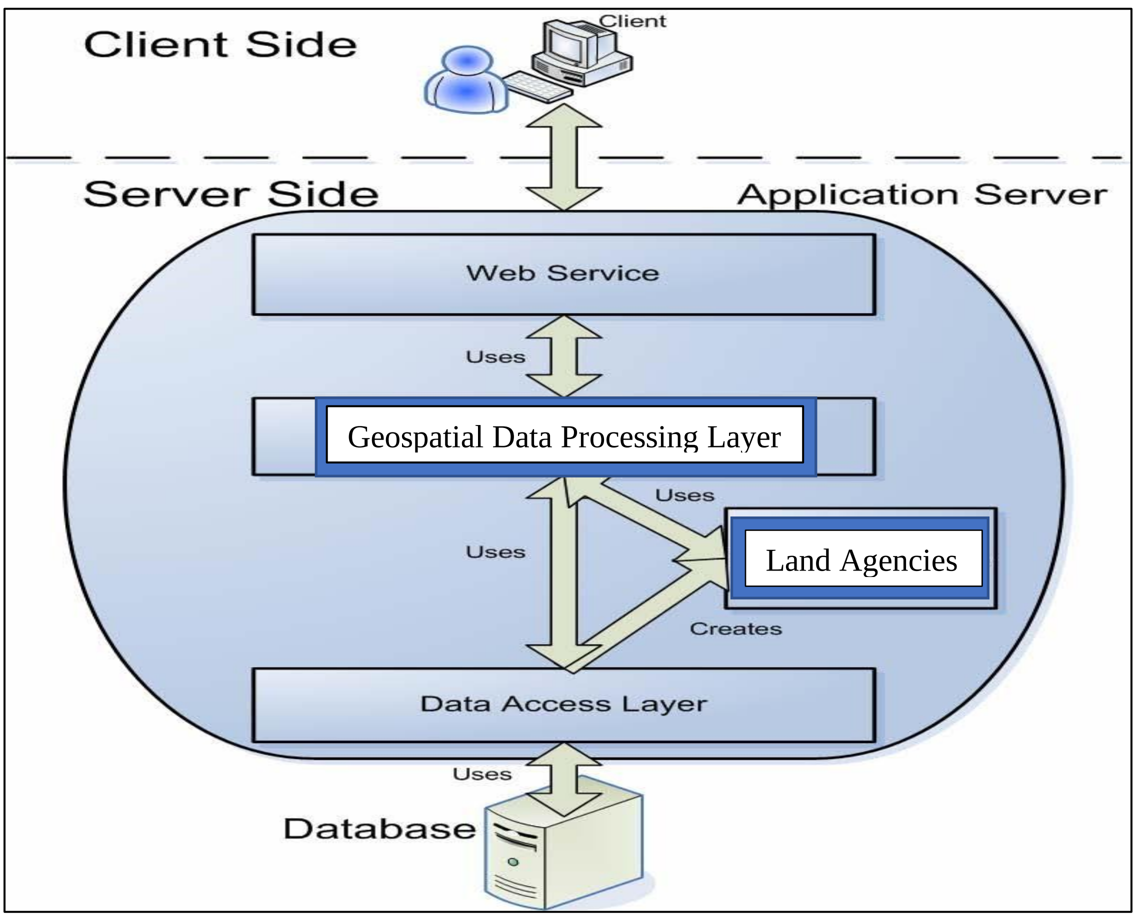


Figure 2. 1 Typical Web Service Architecture (Courtesy, Foster, 2006)

2.4 Cadastral Systems

Surveying and mapping of Kenyan cadastre is unified in the process of land registration; in itself, its development comes after that of land registration. Cadastre development can be grouped into four chronological periods: before 1902, between 1902 and 1945, between 1945 and 1963, and since 1963 (Siriba et al., 2011).

As demonstrated by Wayumba, (2013) for many years, the allocation of Trust Lands was controlled by the Local Authority, the local Provincial Administration and to some extent by the Central Government through the Commissioner of Lands. The process of acquiring land for development has been bureaucratic, slow and expensive. Since the promulgation of the new Constitution and subsequent implementation of the devolved government system, the setting apart procedure will now be carried out at the County level.

According to Thompson et al., (2017) around the word spatial units in a cadastre range from simple two-dimension (2D) parcel units to complex 3D space collections. These units are described at sophistication levels from textural illustrations to absolute, meticulous mathematical interpretations based on measurements and coordinates. However, the two-dimension land parcel is the most familiar spatial unit in a cadastre. It is the primary unit subject to Rights, Restrictions and Responsibilities (RRR) pertaining the cadastre.

LandXML

LandXML is a specialized XML data file format containing civil engineering and survey measurement data commonly used in the Land Development and Transportation Industries. LandXML is an industry standard.

In cadastral data, only geometry is defined in the current way of using LandXML. Topology is dependent on the software reading the data. For instance, encoding of a doughnut shape parcel, one will need to denote both the two polygons and also set out precisely the correlation between these polygons. A doughnut shape parcels are parcels inside parcels that are not connected by boundaries of the same topology. They are two independent parcel rings. An example is an island in the middle of a lake. The island is the inner parcel ring with the lake being the outer parcel ring as illustrated in Figure 2.2

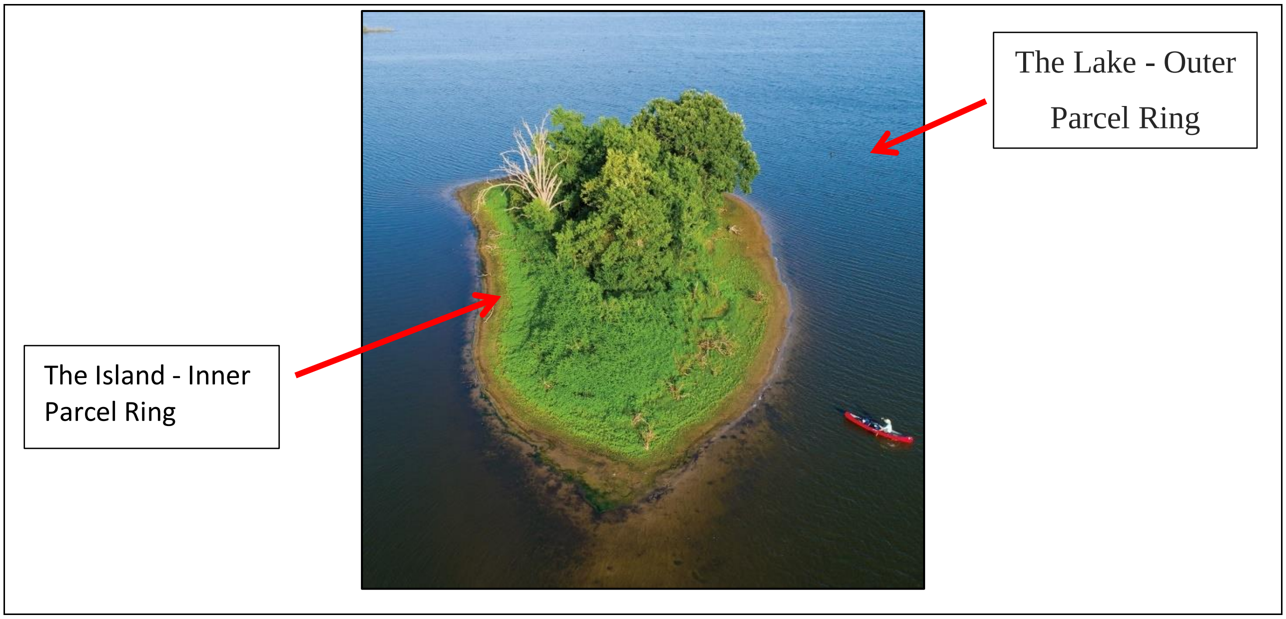


Figure 2. 2 Illustration of a Doughnut Shape Parcel

The software will only need to interpret the syntactic relationship that has been defined in the LandXML as it reads in, regardless of the parcel orientation of the two polygons being the same. Therefore, only a limited number of software can understand this kind of explicit encoding of the relationship. However, in encoding of topology, for example if we set the outer boundary of a parcel as anticlockwise, and the inner one as clockwise, a large number of software can support LandXML as this will align to the universal topology rule for ISO and OGC hence the introduction of InfraGML (Thompson et al., 2017).

Data modeling therefore presents an opportunity to re-engineer the present cadastral systems so as to tackle a wide range of problems. The problem however is that for cadastral modelling to succeed, several adjustments have to be made to existing land administration systems in the developing countries. These include; provision of cadastral information on a uniform geodetic datum, availability of a well-defined and comprehensive coordinate system, availability of fully digitized and georeferenced spatial data, existence of a comprehensive Spatial Data Infrastructure, and a modern administrative set-up and efficient data communication protocol.

2.5 Cadastral maps

According to Siriba et al., (2011) land registration in Kenya is supported by different types of maps. The mainly known maps are the Registry Index Maps representing rural areas and survey plans representing urban areas. RIMs are graphical while the Survey plans are numerical. There are also maps that are only used provisionally to support the process of land registration and they are all graphical. All of these maps are an integral constituent in registration of land. For this reason, they are generated during a first land registration process and in the event of a sub-division of the land, amendment on the maps is done.

2.6 Fixed Boundary Survey Plans

In this study fixed boundary survey plans are of more interest. A survey plan encompasses invisible lines referred to as fixed-boundaries of which through an accurate survey they are defined geometrically. Strictly speaking, the boundary lines are more legal and accurate.

The measurements of all land parcels that have been surveyed and the adjoining/bordering ones shown by the survey plans include: distances, areas, bearings and coordinates. These surveys are based on a designed layout on land parcels of interest hence the name survey plan.

Carrying out of an actual survey is usually preceded by devising a PDP on which the survey is based on. Survey plans are of the highest positional accuracy because they are built on fixed-boundary i.e., they are surveys that are coordinated. Figure 2.3 depicts a sample of a Fixed Boundary Survey Plan.

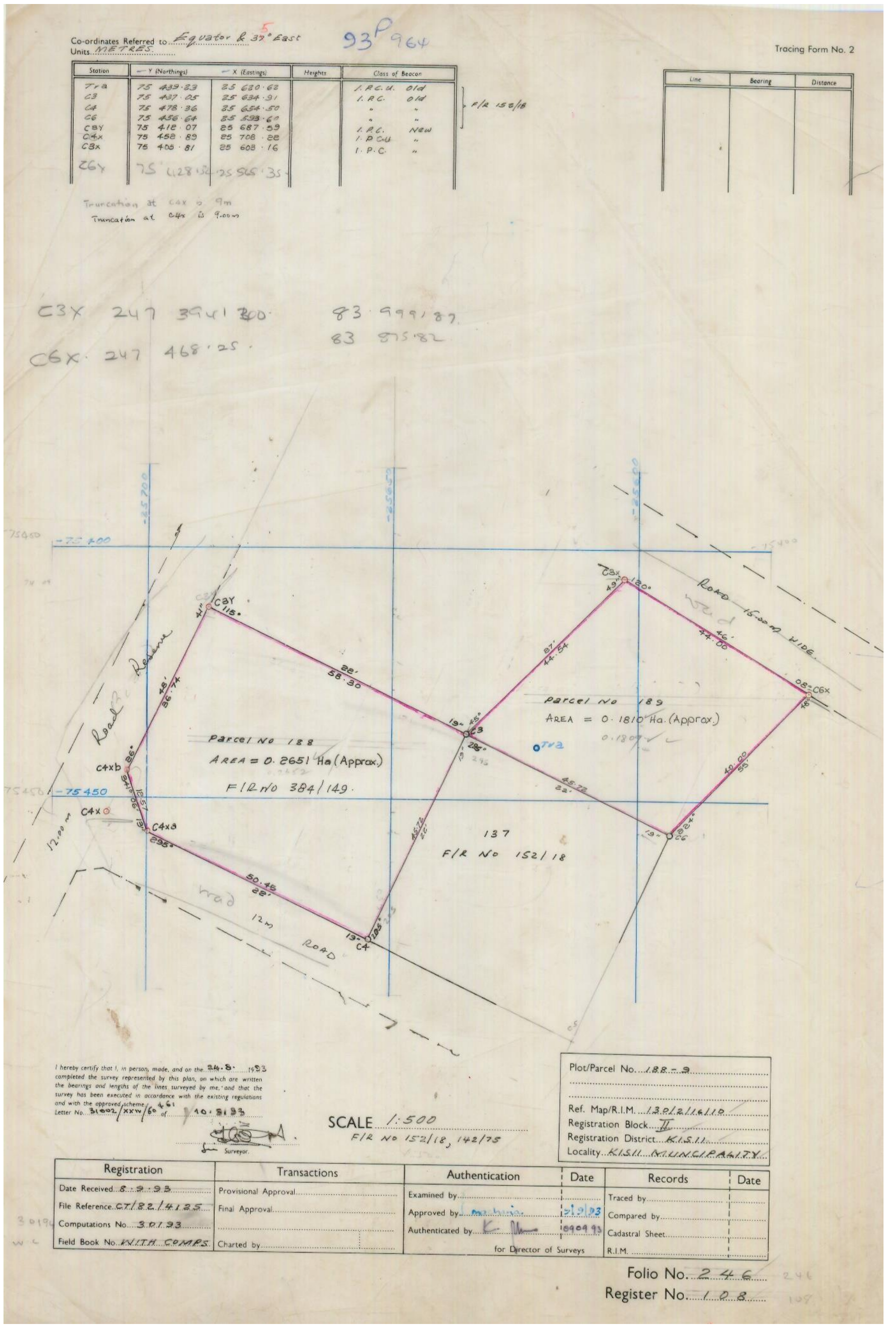


Figure 2. 3 A Sample Survey Plan/F/R (Courtesy, Survey of Kenya)

To ensure that all the requirements and conditions regarding fixed-boundary surveys are met, survey plans are based on methods that are of ground survey. The ground survey methods employ precise techniques and equipment to generate the numeric cadastre from the obtained mathematical data which is accurate and reliable. The traditional optical equipment for survey employed by the methods include: Electro-optical Distance Measurement Equipment (EDM), theodolites, Global

Positioning Systems (GPS), total stations and tacheometers. The main challenge with carrying out fixed boundary surveys is that they are cost intensive (Siriba et al., 2011).

2.7 The Survey Approval Process and LandXML – Land Information New Zealand (LINZ)

As demonstrated by LINZ and PwC Consulting, (2007) in stage one of Landonline New Zealand application, the surveyor lodges a paper cadastral plan and the accompanying documentation to Land Information New Zealand. Information about area cadastral network and land titles is well captured on the survey plan and any surveys in the future are based on it. An elemental capture tool for the plans within the Landonline application, captures electronically a great deal of information obtained from the surveys as represented in the survey plans and stores it as structured data. Scanning of the cadastral plans and other aiding documents is also done to enable storage of the same as images in the image repository of the Landonline application linking them with the acquired information.

The extended Landonline application in stage two allows surveyors to lodge a cadastral survey dataset in a digital form instead of submitting a plan on a paper. Surveyors use any survey software of their choice to enter the dataset. The selected software then produces a Landonline application compatible survey dataset from the survey data. In the Landonline system, format for survey data interchange is based on an XML file. Extensible Markup Language is useful in storing and transferring information between systems and parties. Extensible Markup Language file formats are established from nonproprietary files that are plain text a reason as to why they generally encourage data interchange. Figure 2.4 demonstrates data flow through XML files to and from the application.

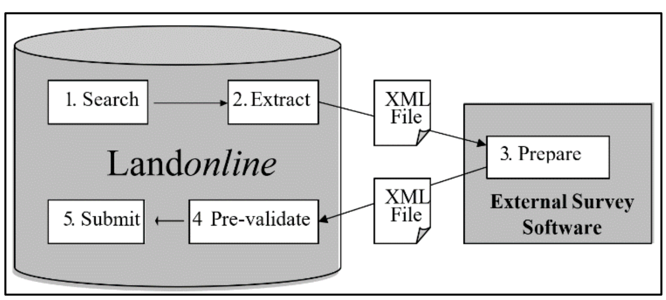


Figure 2. 4 The Application Architecture: Flow of Data (Courtesy, LINZ and PwC Consulting, 2007)

The structure selected for the dataset format has to capture most of the survey information necessary for the Land Information New Zealand application in order to minimize work required to be done by the surveyor after uploading the dataset. LandXML is the transfer data format used by Landonline application. LandXML as an XML data format driven by the industry is considered the best format to ease data interchange during: land planning, land survey processes and civil engineering.

2.8 Survey Capture Tool

With digital survey data lodgement, surveyors are in position to capture datasets in their own chosen software for survey and be able to validate before submitting the dataset with least alteration in the Landonline application. However, incorporating some of the Landonline data prerequisite into their productions could perhaps be a hard task to some vendors of the external software.

In order to achieve flexible data transfer formats requirements, landonline application provides an online tool to facilitate the digital process of submitting survey plan data. The tool enables surveyors to finalize the capture of survey plan datasets that are incomplete when a comparison is done with all the survey data requirements of the Landonline application. The surveyor will be in a position to use the tool to capture the additional cadastral survey information essential to the Landonline application after loading a partly complete dataset, in order for it to correctly process the survey. However, incomplete datasets must still include the attributes and elements of LandXML data format needed by Landonline during import.

2.9 Case Studies

2.9.1 Land Information New Zealand (LINZ)

One of the aims of Land Records Management system of New Zealand is building world class property system.The responsibility of LINZ include: managing land titles, geodetic and cadastral survey systems, topographic information, hydrographic information, managing Crown property and supporting government decision making around foreign ownership.

LINZ provides authoritative seabed and land information including boundary definition and titles for property dealings. Over 60 million records pertaining land are managed by the agency which makes it easy and useful for entities or people interested in land use, buying, selling, managing public land, and location related information. The national land rating system is made consistent and fair by the agency. Transactional services and information to businesses and other government agencies is provided by LINZ. The public can also access geospatial information via the internet through the agency. As a zero-surplus business where there is no loss and profit, LINZ emphasizes on reducing costs at all transactional process points.

2.9.1.1 Landonline Application

According to LINZ and PwC Consulting, (2007) the survey data exchange format for Landonline is based on Extensible Markup Language (XML). As a simple markup language XML can be used in storing and transferring information between systems or parties. Interchange and exchange of data is encouraged in XML since the XML formats are established on nonproprietary plain text files. Landonline application uses LandXML as its transfer format. LandXML (www.LandXML.org) is an industry driven XML format that facilitates exchange and interchange of data during Land Survey, Land Planning and Civil Engineering Processes. A remarkable advancement has been made in incorporating the auxiliary LINZ data requirements in LandXML.

The extended Landonline application allows surveyors to lodge a cadastral survey dataset in a digital form instead of submitting a plan on a paper. Surveyors use any survey software of their choice to enter the dataset. The selected software then produces a Landonline application compatible survey dataset from the survey data.

2.9.1.2 Benefits of the System

Time for processing titles for land buyers has significantly reduced from three days to real time since the introduction of Landonline. The system has also enabled online title searches. The rollout of Landonline led to radical structural change for LINZ as an agency, hence increasing efficiency of the government in public service and also reducing the cost of operation.

The gain to government is certainty of records. Customers receive a higher quality of service when issues of fairness are overcome by 24/7 availability of records to public. Manager of property title regard the physical security of land records as a long-term benefit accruing to the public.

The policy and regulatory functions were separated for the purpose of future outsourcing, and has shifted regulatory responsibilities to professionals. The relationship between government and professionals has become less prescriptive. Monitoring and industry standards are set by professionals.

2.9.2 Kadaster International – Netherlands

As seen in LandMark, (2020) the Netherlands' Cadastre, Land Registry and Mapping Agency (Kadaster), collects and records spatial and administrative data on property and the involved rights. This also goes for telecom networks, aircrafts and ships. In this regard, legal certainty is protected by Kadaster. National mapping and maintenance of the national reference coordinate system is also a responsibility of Kadaster. Furthermore, Kadaster is an advisory for national spatial data infrastructures and land-use issues.

Managing data quality is one of eight goals set out in the five-year strategic plan for The Netherlands Cadastre, Land Registry and Mapping Agency (Kadaster) in 2019. Quality management is also a key part of Kadaster Netherlands' role as one of five Regional Coordinators of EuroRegionalMap (ERM), 1: 250 000 scale topographical mapping produced by EuroGeographics.

Kadaster Netherlands coordinates the annual updating of ERM in its region and merges the data into one dataset for 10 countries: Belarus, Belgium, Czech Republic, Germany, Luxemburg, Poland, Slovakia, Slovenia, Switzerland and The Netherlands. It also performs the Quality Management of ERM for all member countries (LandMark, 2020).

By developing a quality management tool, Kadaster Netherlands has enabled all data producers to validate their ERM data before submitting them to the Regional Coordinators (who use the same tool for validation). This not only saves time and effort, but also improves the quality of the data. Feedback has been positive with some countries adapting the tool to validate their national topographic data.

By enabling the visualisation of spatial data quality in public dashboards for Key Registers in Topography, Large Scale Topography, and Addresses and Buildings, Kadaster Netherlands not only provides data owners and users with a quality indicator, but also the opportunity to contribute to improvements through crowdsourcing. In this way, the agency fulfills its ambition of being a partner in the use of geo-information. Figure 2.5 displays the crowdsourcing statistics in Adresses and Buildings (BAG), Topography (BRT) and Detailed Topography (BGT).

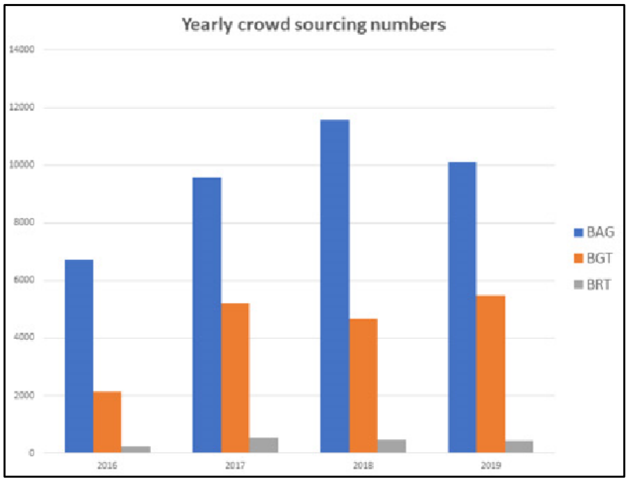


Figure 2. 5 Crowdsourcing Website and Feedback Numbers (Courtesy, LandMark, 2020)

2.9.3 SmartKADASTER - Malaysia

As mentioned by Bin Isa et al., (2015) the information age for cadastral surveying began in a series of separate steps and it is traced back to more than 25 years ago in Malaysia. The Department of Survey and Mapping Malaysia (JUPEM) played an important role in its nationwide implementation success. The cadastral plan is one of the key products of cadastral survey. However, as technology evolve and simplicity made familiar, more effective and better services are expected. This has affected how information on cadastral survey is perceived.

Where enriched cadastral information is needed a paradigm shift becomes necessary. This will provide for multiple usage and allow users to access services based on real cadastral information. On that note, JUPEM ensures that National Digital Cadastral Database is added value by incorporating geospatial information for a multipurpose and smart environment. Interpreted as a tool for decision making, SmartKADASTER encompasses aids of realistic 3D spatial data.

JUPEM developed the SmartKADASTER project with an aim of establishing a SMART and realistic cadastral-based spatial analysis platform to enable: effective decision making, efficiencies, planning and enhanced communication and management to aid SMART functions and services with regard to enabling the Malaysian SMART City. SmartKADASTER development is in phases. Kuala Lumpur and the Federal Territory of Putrajaya are the initial implementation areas of the project. The features of SmartKADASTER are as shown in Figure 2.6

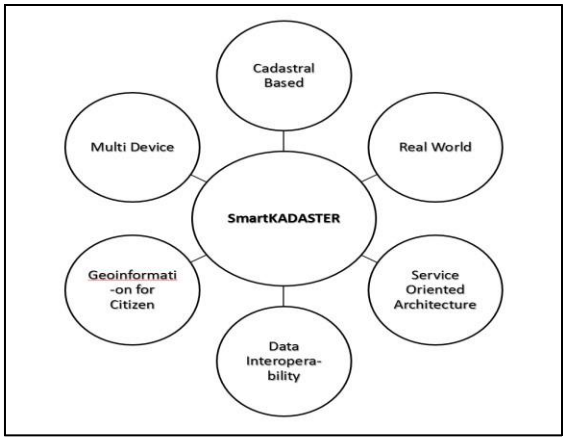


Figure 2. 6 Characteristics of SmartKADASTER (Courtesy, Bin Isa et al., 2015)

SmartKADASTER Interactive Portal (SKIP)

The three-dimension Digital City Modelling of the Federal Territory of Kuala Lumpur and Putrajaya is one of the available SKiP's main products. Taking advantage of 3D datasets produced from available terrestrial 3D survey or airborne technologies led to the development of the 3D City Modelling. This was aimed at meeting the 3D building representation ubiquitous demands and the needed amount of detail was to include facade semantic information not just being limited to geometrics as it were (Bin Isa et al., 2015).

The new capabilities of SmartKADASTER advances the abilities of JUPEM to publish and maintain current survey data while providing a meticulous historic record maintenance of how the data has changed with time. With the aids of other three dimension (3D) spatial data SmartKADASTER is a tool for decision making. As shown by various studies, visual representation of data develops a more captivating experience the user. Tagging of data geospatially enhances location intelligence that enables users to gain new insights and make fast decisions.

SKiP provides a capability of managing simulations such as shadow analysis, best terrain path and rise of water level, based on three-dimension city and terrestrial modelling. Crowd sourcing was also introduced in SKiP, to enable the community to comment and update existing information with additional or new verified information in order to champion collaboration between government and the public.

SKiP also contains functions that enable dissemination and sharing of data via Service Oriented Architecture (SOA). There is also real time access of SmartKADASTER database a function which will enhance legacy application capabilities, complement existing GIS system users and enable development of new applications with accurate and smart spatial analysis abilities.

Figure 2. 7 3D SKiP View of FT Putrajaya (Courtesy, Bin Isa et al., 2015)

2.10 The Cadastral and Modern Land Administration System of Kenya

Technology enhances: collection, storage, administration and management of land information and this is a characteristic of an organized society. In other words, technology presents capacity by making use of location as the starting point for spatial enablement in land information and cadastral systems integration.

Furthermore, an operational SDI is essential in the development of an effective land management system for sustainable development. The most important aspects of a cadastre include: automation, integration of the functions of land administration, collaboration on common standards and data modeling, accessibility of information pertaining the cadastre and quality of information.

The Land Information System has evolved to like the one in E-Citizen portal whereby in order for one to conduct searches the individual makes a request through an application and has to wait for at least 24hrs to get a feedback. This means at the back-office operations are carried out manually before the feedback is scripted. To apply for a land search the user would comply with the requirements as depicted in Figure 2.8

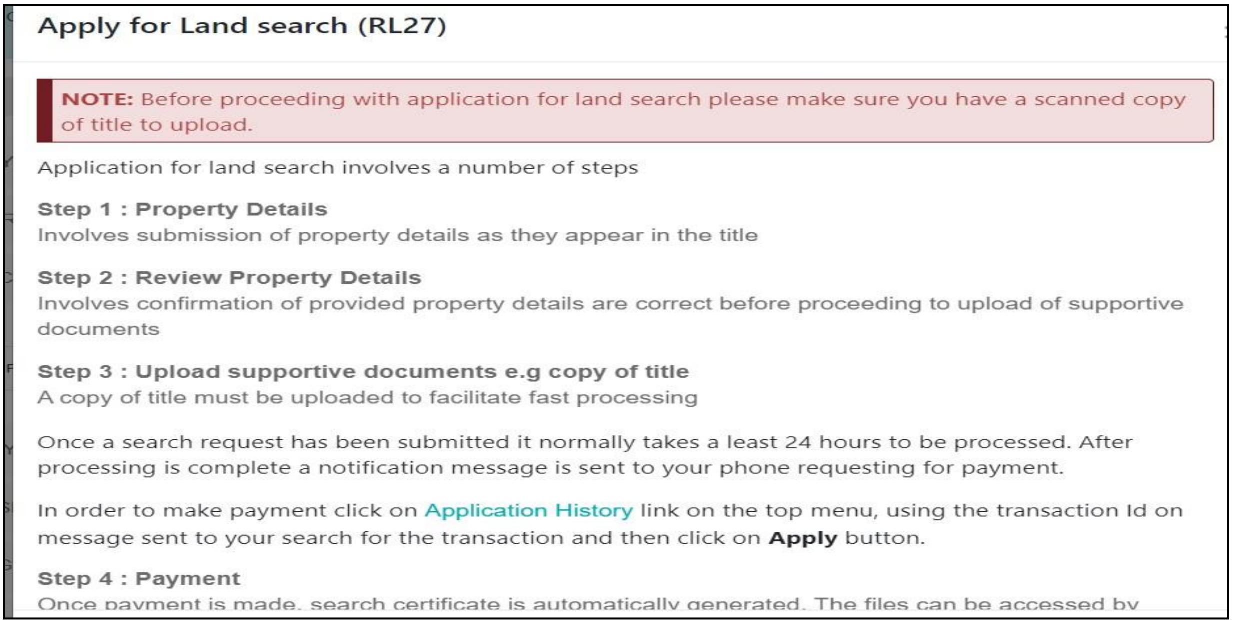


Figure 2. 8 Application for Land Search Steps

The Ministry of Land and Physical Planning (MoLPP) recently developed a web-based National Land Information Management System named Ardhisasa. Ardhisasa is an online platform that allows citizens, stakeholders and interested parties to interact with land information and processes undertaken by Government. It was developed through the joint efforts by the Ministry of Lands and Physical Planning (MoLPP) and also the National Land Commission (NLC) and key stakeholders in Government. The applicant first of all registers their credentials in the system before they can log in for the land services. The LIMS home and service pages are as shown in figure 2.9 and 2.10 respectively.

Figure 2. 9 Ardhisasa Homepage

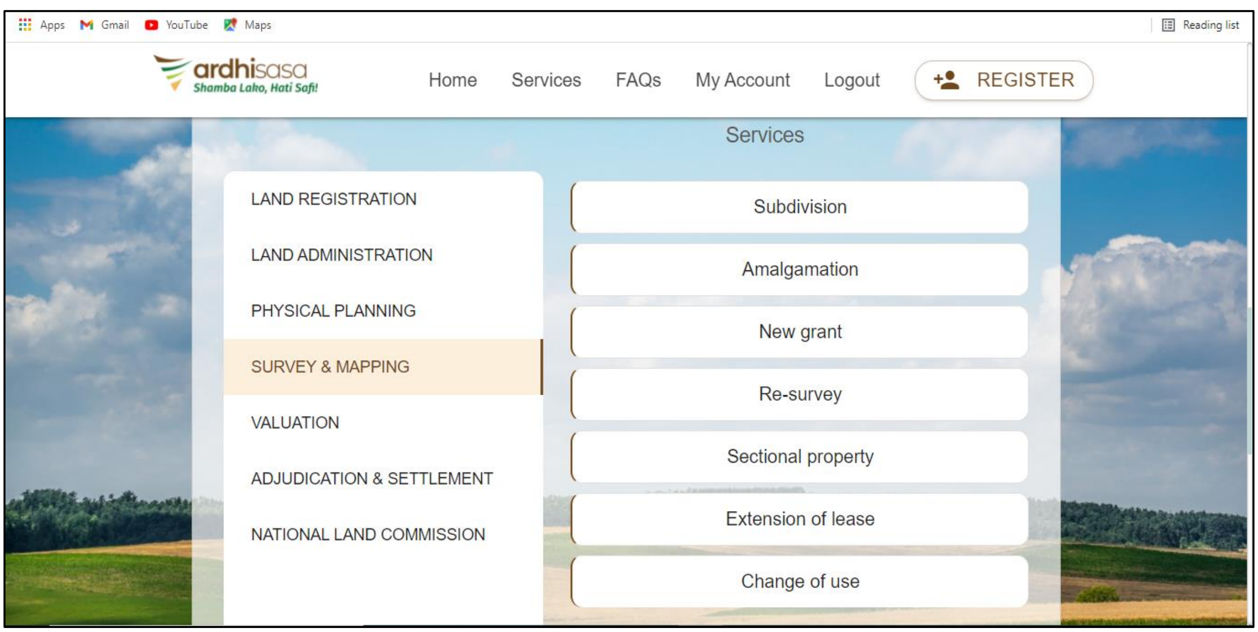


Figure 2. 10 Ardhisasa Services

The realization of Ardhisasa NLIMS is a great achievement by the Kenya Ministry of Land and Physical Planning (MoLPP). It is a milestone towards realization of a modern cadastre.

2.11 Land Administration Domain Model

According to Siriba et al., (2011) Kenyan cadastral system does not conform to any common data model since no formalized cadastral and land administration data model is in existence. The Land Administration Domain Model (ISO, 2008) and Cadastre 2014 which are efforts made internationally towards data modeling, could be the initial points for implementing a national land management and administration data model.

Lemmen et al., (2015) mentioned that, ISO TC211 19152:2012 provided a constant source of inspiration for the LandInfra LandDivision and Condominium RCs. However, it should be noted that the scope of LandInfra in this area is much more limited to land development and civil engineering infrastructure facilities. The emphasis of LandInfra on surveying and infrastructure indicates minimizing entirely the legal administrative aspects of land development. This is attained by modelling what is needed to account for the surveying affiliated activities, including definition of the legal entities, measured boundary, as well as identification of the signing parties.

The primary concern for the initial release of LandInfra is the determination of land ownership as it relates to infrastructure facilities. For example, does an owner have enough road right of way to construct a road or does he/she need to acquire additional property (or easements) from adjacent property owners? The legal/rights issues addressed by LADM are minimized in LandInfra to what is needed to know to define and measure the legal boundary.

Similarly, the recording issue is minimized, because the broad diversity of implementations makes it difficult to achieve a consistent and understandable world-wide specification of cadastral recording. These do provide areas for future LandInfra development. A more detailed comparison between LandInfra and LADM is provided as LandInfra Annex D.2.

Reliance on ICT is the global trend in implementation of land administration systems. According to the fourth statement of cadastre 2014, the use of pen and paper will be substituted by computer systems that enable automation in cadastral mapping. Even though, there is evidence in some departmental sections responsible for the cadastre of automating some processes, the cadastral system is still analogue. Moreover, capturing data in the field employs digital land surveying equipment but submission of the cadastral surveys to the Director of Surveys for approval is still vital to be done in an analogue way. Taking into account common standards and modeling of cadastral data, the cadastre along with the land register have to be automated if we are to talk of modern technology-based land administration system.

As demonstrated by Mwungu, (2020) system development consists of three concatenated stages which include analysis, design and implementation. System scope and behaviour are system-specific whereas the scope of modelling information is usually enterprise-wide.

Design and implementation of systems mainly focuses on mapping the conceptual model of analysis onto design and runtime models, respectively. A conceptual model is the result of analysis and it is commonly used to produce the design model in the design process. Implementation, contrastingly, maps the design model such as object-oriented programs, relational databases onto runtime architectures, often referred to as the runtime model. The system analyst usually establishes subject domain concepts including attributes, entities and their interrelationships, operations and identifiers. In LIMS, the “things” that are involved in providing land information are the subject domain concepts. Some of the entities are tangible, such as a person and a land parcel; but others are more abstract, such as servitude and tenure (Mwungu, 2020).

2.12 LandXML, LandInfra and InfraGML

Following evaluation of the LandXML 1.2 schema, the Open Geospatial Consortium Land and Infrastructure Domain Working Group (LandInfraDWG) put forward the advancement of an alternative standard to be part of the OGC standards inception. The Preface of OGC-Consortium Model 15-111 describes the merits and approach used by the OGC Land and Infrastructure Standards Working Group (LandInfraSWG). With shared interest by the buildingSMART International Infrastructure Room, it was agreed that OGC-Consortium Model 15-111 would just be a concepts document. IFC, GML and possibly other encodings would follow as separate efforts of standardization. An anticipated GML encoding will be compatible with other GML standards such as CityGML.

As demonstrated by Open Geospatial Consortium, (2017) OGC InfraGML Encoding Standard exhibits the implementation dependent GML encoding of land and civil engineering infrastructure facilities supporting concepts stipulated in the OGC Land and Infrastructure Conceptual Model Standard. The Conceptual model subject areas include land division, railway, road, facilities, land features, projects, survey (encompassing survey results, equipment used and observations), alignment and condominiums. InfraGML is published as a multi-part standard.

LandInfra an open standard which is relatively new is a standard for representing and modelling land and infrastructure features. Initially from the spatial sphere, LandInfra was in the recent past proposed to succeed LandXML (Kumar et al., 2019). LandXML is an open data model based on XML developed to represent survey measurement data and civil engineering. However, it not being recognised officially as a standard by any organization involved with standards such as ISO or OGC, caused a confusion concerning the standard's future in the marketplace.

In order to align LandXML with the OGC standards, a LandGML Interoperability Experiment was inaugurated in 2004 by OGC to make it amenable with the geospatial data GML standard. Ensuing this effort in 2013, LandInfra Standards Working Group (LandInfra SWG) made a review on LandXML and put efforts on determining how to carry own with its support in the best manner possible to the existing users.

Several problems discovered in LandXML 1.2 were documented. Further, there is no underlying conceptual model, requirements definition, user guide or formal published documentation of LandXML. Therefore, LandInfra a new OGC standard was developed gleaned from a subset of functionality from LandXML but implemented with GML as InfraGML and supported by a UML conceptual model.

LandInfra consorts the surveying needs to locate infrastructure facilities on the terrain and encompasses both subsurface and topography information in acquiescence with interests in land. It thus includes but no limited to survey, land features, land division elicited from administration, buildings, roads, railways, districts and jurisdictions. LandInfra also in cooperates interests in land such as land parcels, condominiums and easements. All these in civil engineering and land infrastructure facilities.

Having a common underlying Conceptual Model across all LandInfra encodings will help insure the compatibility across multiple encoding standards. The goal of OGC-Consortium Model 15-111 was therefore to initiate and develop a documentation of a common set of concepts that constitute land and civil engineering infrastructure applications.

CityGML

CityGML is a conventional semantic information model for 3D urban objects representation that can be shared over different programs and applications (Gröger et al., 2012). CityGML as an XML based format and an open data model is designed for storage and interchange of virtual 3D city models. Its implementation is as a Geography Markup Language 3 (GML3) application schema of the expandable international standard for geospatial data encoding and interchange issued by the ISO TC211 and the Open Geospatial Consortium (OGC). CityGML comprises numerous standards from the W3C Consortium, the ISO 191xx family, the Web 3D Consortium, OASIS and the Open Geospatial Consortium.

The motivation behind its development is the rising number of companies and cities that are building virtual 3D city models for different areas of application such as disaster management, urban planning, 3D cadaster, mobile telecommunication, facility management, vehicle and pedestrian navigation, tourism and environmental simulations.

This study therefore explores the practicality of developing an XML standard for survey data interchange to provide for rules and guidelines that will enable a surveyor to topologically encode two-dimensional (2D) land parcel units in InfraGML.

The method in suggestion employs a mixed dimensional topological structure form that encompasses spatial land units that are simple and effective in space requirements in boundary definitions sharing. This provides 2D digital cadastral geodatabase with primary data that minimizes inconsistency and redundancy. This approach builds up a conceptual model that can be extended to the land information system geodatabase itself, including historical record of the spatial unit structure (pedigree) maintenance.

2.13 Digitization

Common Coordinate Geometry (COGO) Workflows

(i) ArcGIS – ArcMap

According to Esri, the COGO functionality together with the COGO toolbar in ArcMap allows for creation and maintenance of surveyed features and land parcels in a geodatabase. ArcInfo Workstation also comprises of other Esri functionality that provides much the same capabilities such as COGO extension and land parcel editing. Features of high accuracy are guaranteed when COGO methods are used in creating parcel boundaries from a cadastral plan.

Adding Parcels from a Survey Plan - Creating the External Boundaries of the Parcels

Creation of the parcel's external boundaries is the initial stage. These external boundaries can be entered into a line layer by the use of the traverse window. To achieve hive accuracy in creation of the external boundaries, one needs to start their traverse at a boundary or an existing ground control point.

Figure 2. 11 External Boundaries of the Parcel (Courtesy, Esri)

Creating Internal Parcel Boundaries

After creation of the external boundaries the user's next step will be adding the internal boundaries of the respective parcels. The user is required to digitize boundary if one divides back lot boundary and each road. One has an alternative also of entering a start point, the direction and distance of the boundary using the 2 Point Line window. This ensures that the new line features are stored with the COGO description entered. To ensure that boundaries snap one ought to set the snapping environment before using either technique.

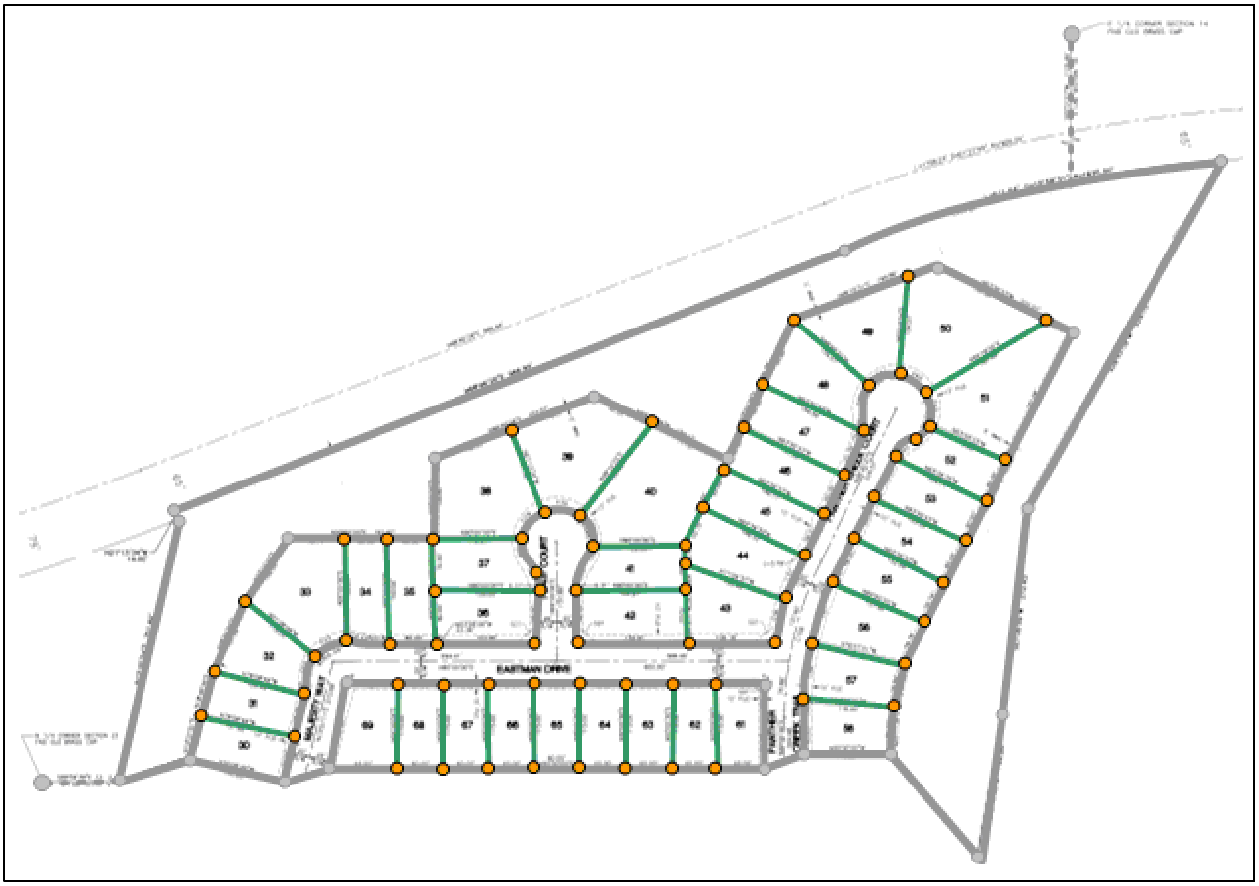


Figure 2. 9 Internal Parcel Boundaries (Courtesy, Esri)

(ii) **Quantum GIS**

Coordinate Geometry (CoGo) Plugin

Cadastral surveying encompasses demarcation and the survey of land for the purpose of land parcels definition for registration in a land registry. Any captured survey in a Geographic Information System is a true reflection of what is represented by survey diagrams and what is on the ground. As demonstrated by Admire Nyakudya, (2018) CoGo as a tool provides for; capturing and editing of beacons, definition of distances and bearings thus defining the land parcels automatically. Multiple users working on the same PostgreSQL/PostGIS database are also supported by the plugin. It facilitates accurate and efficient capturing of data as well as bulk uploading of structured data by operators. Figure 2.10 depicts some of the functionalities.

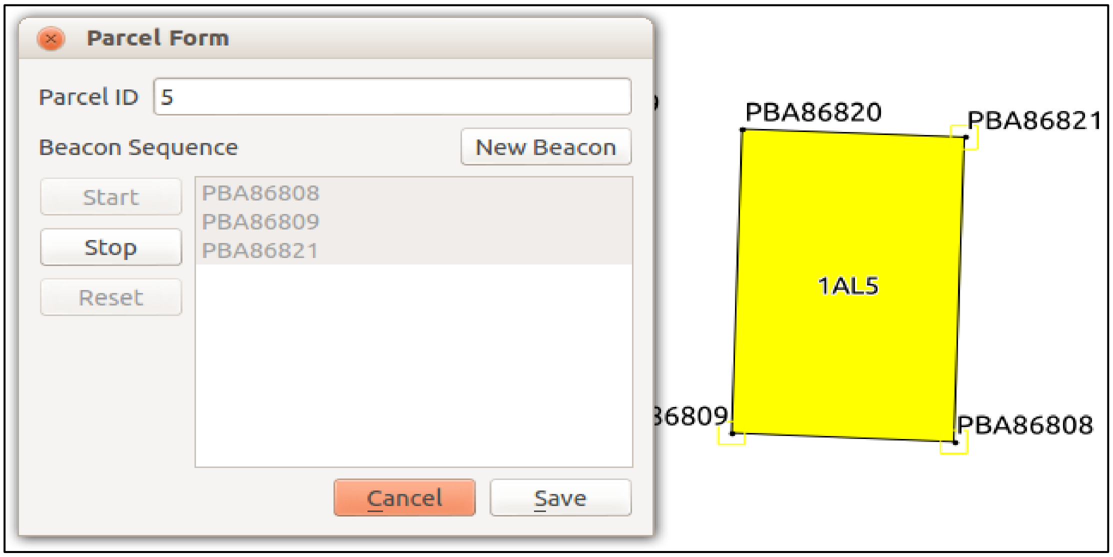


Figure 2. 10 Entry of Land Parcels Beacon Coordinates (Courtesy, Admire Nyakudya, 2018)

2.14 Conceptual Framework

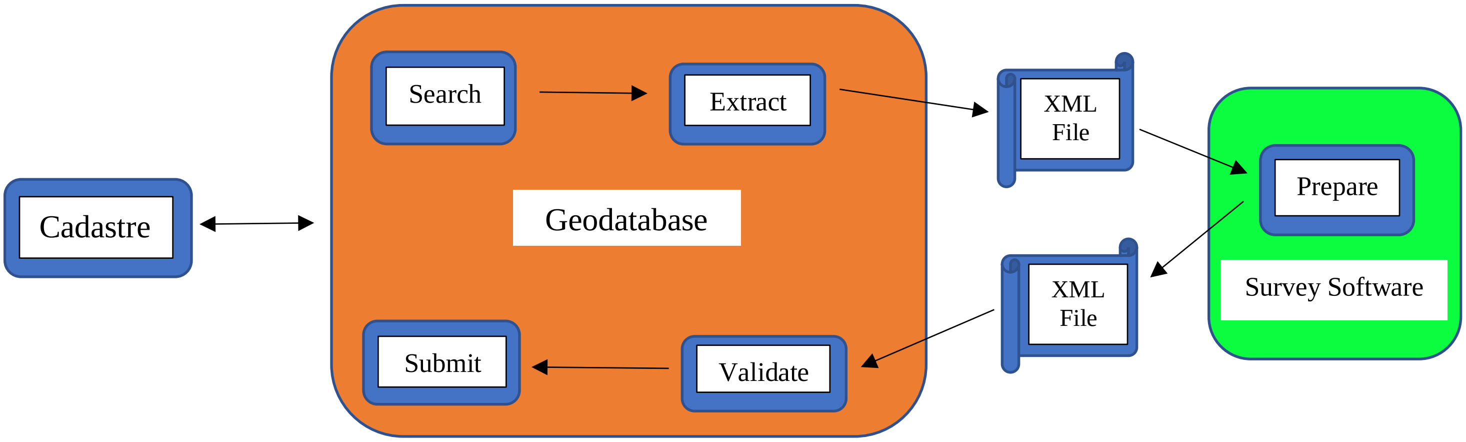


Figure 2. 12 Conceptual Framework (Courtesy, Researcher)

CHAPTER 3: MATERIALS AND METHODS

This section involves representation of the Kenyan cadastral data in an approach to enable both geometrical and topological encoding schemes in the development of the XML Framework.

The survey plans concept and the exploration on topology for data pertaining the cadastre are discussed. Description of a selection of two-dimensional (2D) real and theoretical cases was also done. It entails a conceptual encoding and demonstration of the actual encoding in InfraGML.

Ten fixed boundary survey plans were used in identifying key data elements of a cadastral plan used in encoding survey data in the InfraGML Standard. The encoding standard was developed in line with the standards set by OGC Land and Infrastructure Standards Working Group (LandInfraSWG).

Programs were written to provide these functionalities: encoding of the plan in an easy way, writing of the two-dimensional spatial units of the plans to a PostGIS/PostgreSQL database, translating the simplified spatial units into InfraGML to allow interchange of cadastral data with established GIS programs such as ArcGIS and QGIS. Figure 3.1 depicts a flow diagram illustrating the fundamental processes.

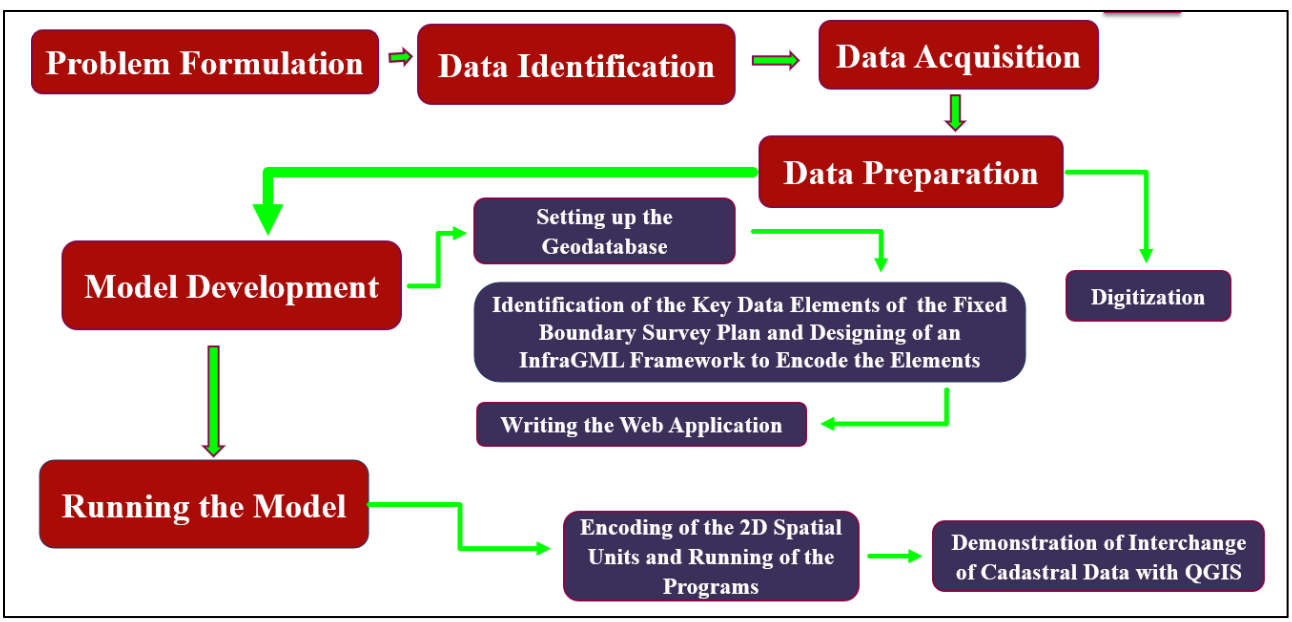


Figure 3. 1 Methodology Flow Diagram

3.1 Data Identification and Acquisition

Fixed boundary survey plans used in the study were identified and acquired from Survey of Kenya. They are ten in number and include:

(i) FR/No 246/108

(ii) FR/No 46/9

(iii) FR/No 140/107

(iv) FR/No 225/51

(v) FR/No 73/196

(vi) FR/No 131/74

(vii) FR/No 111/96

(viii) FR/No 138/154

(ix) FR/No 394/20

(x) FR/No 46/58

3.2 Data Preparation

3.2.1 Digitization

Digitization of the Fixed Boundary Survey Plans was done using the Coordinate Geometry (CoGo) plugin in QGIS. The CoGo plugin is capable of handling both types of coordinates used in land surveying; polar coordinates (bearing and distance) and cartesian coordinates (x,y; longitude, latitude). Figure 3.2 depicts digitization of parcel no 189 of F/R no 246/108. Digitization of the survey plans yielded parcels as seen in Figure 3.3.

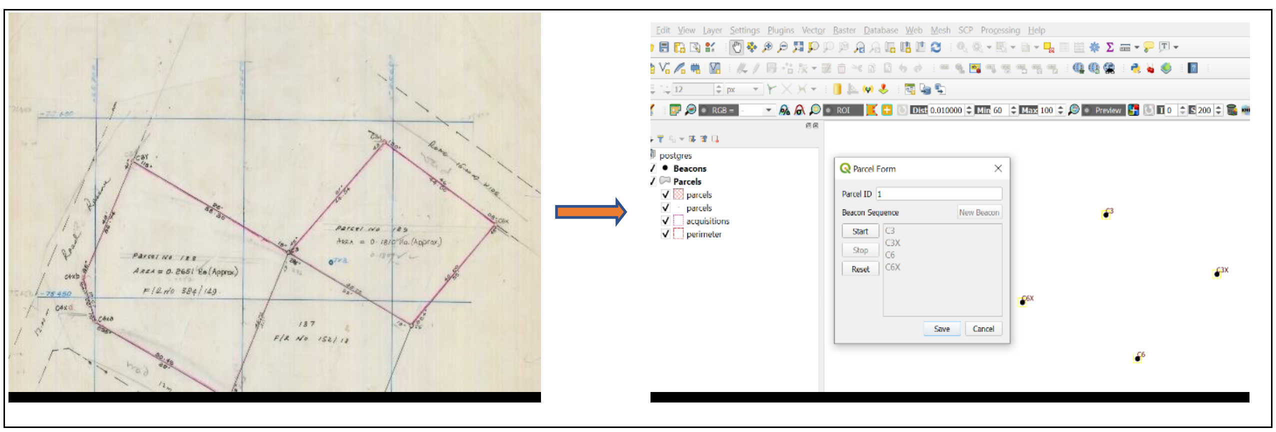


Figure 3. 2 Digitization of Parcel No. 189 of F/R No. 246/108

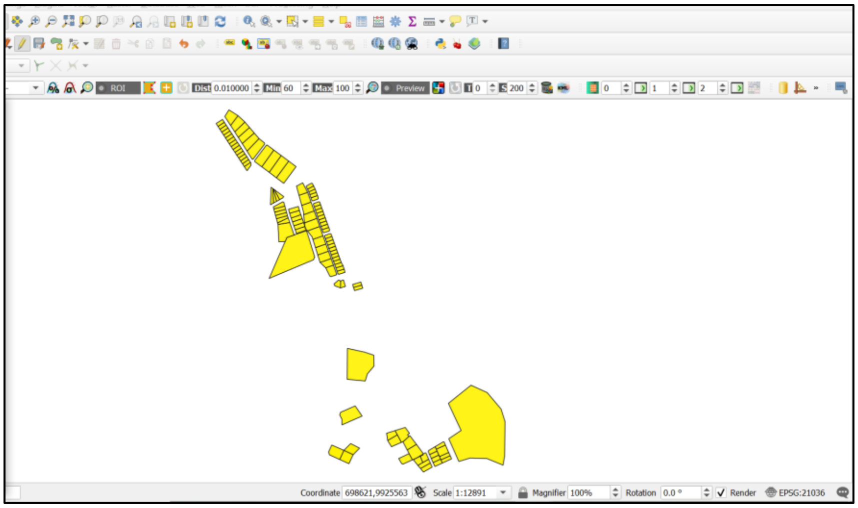


Figure 3. 3 The Resultant Parcels after Digitization

3.3 Model Development

3.3.1 Setting up the Geodatabase

The study used PostgreSQL/PostGIS database management system - PostGIS is a PostgreSQL object-relational database's spatial extender. It provides support to geographic objects allowing running of location queries in SQL. The choice of the database was also influenced by the fact that it is powerful, open source and can handle xml data well.

3.3.2 Identification of the Survey Plan key data elements to be encoded in the InfraGML Framework

Ten fixed boundary survey plans were used in identifying key data elements of a cadastral plan used in encoding survey data in the InfraGML Framework.

3.3.3 Development of the InfraGML Encoding Standard

The development of the framework was guided by the alignment of XML elements which are in accordance with the Geography Markup Language XML Schema Definition (GML XSD) specified in InfraGML schemas. GML XSD provides the syntax and defines a way to represent attributes and elements in an XML document, in other words, it is a way to describe the structure of an XML document.

The encoding in InfraGML followed some principles outlined in Land and Infrastructure Standards Working Group of OGC (LandInfraSWG) in OGC-Consortium Model 15-111. The encoding schema is as shown in Figure 3.4.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.1.1">
    <annotation>
        <documentation>GML Subset schema for gml:metaDataProperty,gml:Polygon,gml:Point,gml:LineString,gml:LinearRing,
        gml:exterior,gml:interior,gml:surfaceProperty,gml:multiSurfacePropertyType,gml:_Feature,gml:featureMember,
        gml:CodeListType,gml:AngleType,gml:MeasureListType,gml:centerOf,gml:extentOf,gml:_MetaData,gml:Observation,
        gml:validTime,gml:beginPosition,gml:endPosition,gml:TimePeriod,gml:multiExtentOf,gml:_FeatureCollection,
        gml:CircleByCenterPoint,gml:Curve, written by gmlSubset.xslt.
        </documentation>
    </annotation>
    <import namespace="http://www.w3.org/1999/xlink" schemaLocation="../../survey/1.0.0/survey.xsd"/>
    <!-- ================================================= -->
    <element name="metaDataProperty" type="gml:MetaDataPropertyType">
        <annotation>
            <documentation>Refers to a package of metadata that contains metadata properties.</documentation>
        </annotation>
    </element>
    <!-- ================================================= -->
    <complexType name="MetaDataPropertyType">
        <annotation>
            <documentation>Base type for complex metadata property types.</documentation>
        </annotation>
        <sequence minOccurs="0">
            <any processContents="lax"/>
            <!-- <element ref="gml:_MetaData"/> -->
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
        <attribute name="about" type="anyURI" use="optional"/>
    </complexType>
    <!-- ================================================= -->
    <attributeGroup name="AssociationAttributeGroup">
        <annotation>
            <documentation>Attribute group used to enable property elements to refer to their value remotely.
            It contains the simple link components from xlinks.xsd,
            with all members optional, and the remoteSchema attribute, which is also optional.
            These attributes can be attached to any element, thus allowing it to act as a pointer.
            The 'remoteSchema' attribute allows an element  that carries link attributes to indicate that the element
            is declared  in a remote schema rather than by the schema that constrains the current document instance.
            </documentation>
        </annotation>
        <attributeGroup ref="xlink:simpleLink"/>
        <attribute ref="gml:remoteSchema" use="optional"/>
    </attributeGroup>
    <!-- ================================================= -->
    <attribute name="remoteSchema" type="anyURI">
        <annotation>
            <documentation>Reference to an XML Schema fragment that specifies the content model of the propertys value.
            This is in conformance with the XML Schema Referencing Schemas from Elsewhere.
            </documentation>
        </annotation>
    </attribute>
    <!-- ================================================= -->
    <element name="Polygon" type="gml:PolygonType" substitutionGroup="gml:_Surface"/>
    <!-- ================================================= -->
    <complexType name="PolygonType">
        <annotation>
            <documentation>A Polygon is a special surface that is defined by a single surface patch. The boundary of this patch is
            coplanar and the polygon uses planar interpolation in its interior. It is backwards compatible with the Polygon of GML 2,
            GM_Polygon of ISO 19107 is implemented by PolygonPatch.
            </documentation>
        </annotation>
        <complexContent>
            <extension base="gml:AbstractSurfaceType">
                <sequence>
                    <element ref="gml:exterior" minOccurs="0"/>
                    <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!-- ================================================= -->
    <complexType name="AbstractSurfaceType">
        <annotation>
            <documentation>An abstraction of a surface to support the different levels of complexity. A surface is always a continuous
            region of a plane.
            </documentation>
        </annotation>
        <complexContent>
            <extension base="gml:AbstractGeometricPrimitiveType"/>
        </complexContent>
    </complexType>
    <!-- ================================================= -->
    <complexType name="AbstractGeometricPrimitiveType" abstract="true">
```

```xml
		<annotation>
			<documentation>This is the abstract root type of the geometric primitives. A geometric primitive is a geometric object that
			is not decomposed further into other primitives in the system. All primitives are oriented in the direction implied by the
			sequence of their coordinate tuples.
			</documentation>
		</annotation>
		<complexContent>
			<extension base="gml:AbstractGeometryType"/>
		</complexContent>
	</complexType>
	<!-- ================================================== -->
	<complexType name="AbstractGeometryType" abstract="true">
		<annotation>
			<documentation>All geometry elements are derived directly or indirectly from this abstract supertype. A geometry element may
			have an identifying attribute ("gml:id"), a name (attribute "name") and a description (attribute "description"). It may be
			associated with a spatial reference system (attribute "srsName"). The following rules shall be adhered: - Every geometry
			type shall derive from this abstract type. - Every geometry element (i.e. an element of a geometry type) shall be directly
			or indirectly in the substitution group of _Geometry.
			</documentation>
		</annotation>
		<complexContent>
			<extension base="gml:AbstractGMLType">
				<attribute name="gid" type="string" use="optional">
					<annotation>
						<documentation>This attribute is included for backward compatibility with GML 2 and is deprecated with GML 3.
						This identifer is superceded by "gml:id" inherited from AbstractGMLType. The attribute "gid" should not be used
						anymore and may be deleted in future versions of GML without further notice.
						</documentation>
					</annotation>
				</attribute>
				<attributeGroup ref="gml:SRSReferenceGroup"/>
			</extension>
		</complexContent>
	</complexType>
	<!-- ================================================== -->
	<complexType name="AbstractGMLType" abstract="true">
		<annotation>
			<documentation>All complexContent GML elements are directly or indirectly derived from this abstract supertype
	to establish a hierarchy of GML types that may be distinguished from other XML types by their ancestry.
	Elements in this hierarchy may have an ID and are thus referenceable.</documentation>
		</annotation>
		<sequence>
			<group ref="gml:StandardObjectProperties"/>
		</sequence>
		<attribute ref="gml:id" use="optional"/>
	</complexType>
	<!-- ================================================== -->
	<group name="StandardObjectProperties">
		<annotation>
			<documentation>This content model group makes it easier to construct types that
	  derive from AbstractGMLType and its descendents "by restriction".
	  A reference to the group saves having to enumerate the standard object properties.
			</documentation>
		</annotation>
		<sequence>
			<element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
			<element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
			<element ref="gml:description" minOccurs="0"/>
			<element ref="gml:name" minOccurs="0" maxOccurs="unbounded">
				<annotation>
					<documentation>Multiple names may be provided.  These will often be distinguished by being assigned by different
					authorities, as indicated by the value of the codeSpace attribute.  In an instance document there will usually
					only be one name per authority.
					</documentation>
				</annotation>
			</element>
		</sequence>
	</group>
	<!-- ================================================== -->
	<element name="description" type="gml:StringOrRefType">
		<annotation>
			<documentation>Contains a simple text description of the object, or refers to an external description.
			</documentation>
		</annotation>
	</element>
	<!-- ================================================== -->
	<complexType name="StringOrRefType">
		<annotation>
			<documentation>This type is available wherever there is a need for a "text" type property. It is of string type,
			so the text can be included inline, but the value can also be referenced remotely via xlinks from the
			AssociationAttributeGroup. If the remote reference is present, then the value obtained by traversing the link should be used,
			and the string content of the element can be used for an annotation.
			</documentation>
		</annotation>
		<simpleContent>
```

```xml
			<extension base="string">
				<attributeGroup ref="gml:AssociationAttributeGroup"/>
			</extension>
		</simpleContent>
	</complexType>
	<!-- ================================================== -->
	<!-- ================================================== -->
	<element name="name" type="gml:CodeType">
		<annotation>
			<documentation>Label for the object, normally a descriptive name. An object may have several names,
			typically assigned by different authorities.  The authority for a name is indicated by the value of its (optional)
			codeSpace attribute.  The name may or may not be unique, as determined by the rules of the organization responsible
			for the codeSpace.
			</documentation>
		</annotation>
	</element>
	<!-- ================================================== -->
	<complexType name="CodeType">
		<annotation>
			<documentation>Name or code with an (optional) authority.  Text token. If the codeSpace attribute is present, then its
			value should identify a dictionary, thesaurus or authority for the term, such as the organisation who assigned the value, or
			the dictionary from which it is taken. A text string with an optional codeSpace attribute.
			</documentation>
		</annotation>
		<simpleContent>
			<extension base="string">
				<attribute name="codeSpace" type="anyURI" use="optional"/>
			</extension>
		</simpleContent>
	</complexType>
	<!-- ================================================== -->
	<attribute name="id" type="ID">
		<annotation>
			<documentation>Database handle for the object.  It is of XML type ID, so is constrained to be unique in the XML document within which
			it occurs.  An external identifier for the object in the form of a URI may be constructed using standard XML and XPointer methods.
			This is done by concatenating the URI for the document, a fragment separator, and the value of the id attribute.
			</documentation>
		</annotation>
	</attribute>
	<!-- ================================================== -->
	<attributeGroup name="SRSReferenceGroup">
		<annotation>
			<documentation>Optional reference to the CRS used by this geometry, with optional additional information to simplify use when
			a more complete definition of the CRS is not needed.
			</documentation>
		</annotation>
		<attribute name="srsName" type="anyURI" use="optional">
			<annotation>
				<documentation>For well known references it is not required that the CRS description exists at the
				location the URI points to. If no srsName attribute is given, the CRS must be specified as part of the larger context this
				geometry element is part of, e.g. a geometric element like point, curve, etc. It is expected that this attribute will be specified
				at the direct position level only in rare cases.
				</documentation>
			</annotation>
		</attribute>
		<attribute name="srsDimension" type="positiveInteger" use="optional">
			<annotation>
				<documentation>The "srsDimension" is the length of coordinate sequence (the number of entries in the list). This dimension is
				specified by the coordinate reference system. When the srsName attribute is omitted, this attribute shall be omitted.
				</documentation>
			</annotation>
		</attribute>
		<attributeGroup ref="gml:SRSInformationGroup"/>
	</attributeGroup>
	<!-- ================================================== -->
	<!-- ================================================== -->
	<attributeGroup name="SRSInformationGroup">
		<annotation>
			<documentation>Optional additional and redundant information for a CRS to simplify use when a more complete definition of the
			CRS is not needed. This information shall be the same as included in the more complete definition of the CRS, referenced by the
			srsName attribute. When the srsName attribute is included, either both or neither of the axisLabels and uomLabels attributes
			shall be included. When the srsName attribute is omitted, both of these attributes shall be omitted.
			</documentation>
		</annotation>
		<attribute name="axisLabels" type="gml:NCNameList" use="optional">
			<annotation>
				<documentation>Ordered list of labels for all the axes of this CRS. The gml:axisAbbrev value should be used for these axis
				labels, after spaces and forbiddden characters are removed. When the srsName attribute is included, this attribute is optional.
				When the srsName attribute is omitted, this attribute shall also be omitted.
				</documentation>
			</annotation>
		</attribute>
		<attribute name="uomLabels" type="gml:NCNameList" use="optional">
			<annotation>
				<documentation>Ordered list of unit of measure (uom) labels for all the axes of this CRS. The value of the string in the
				gml:catalogSymbol should be used for this uom labels, after spaces and forbiddden characters are removed. When the
				axisLabels attribute is included, this attribute shall also be included. When the axisLabels attribute is omitted, this attribute
				shall also be omitted.
				</documentation>
			</annotation>
		</attribute>
	</attributeGroup>
	<!-- ================================================== -->
	<simpleType name="NCNameList">
		<annotation>
			<documentation>A set of values, representing a list of token with the lexical value space of NCName. The tokens are seperated by whitespac
		</annotation>
		<list itemType="NCName"/>
	</simpleType>
```

```xml
    <!-- ================================================== -->
    <!-- ================================================== -->
    <element name="exterior" type="gml:AbstractRingPropertyType">
        <annotation>
            <documentation>A boundary of a surface consists of a number of rings. In the normal 2D case, one of these rings is distinguished
            as being the exterior boundary. In a general manifold this is not always possible, in which case all boundaries shall be listed as
            interior boundaries, and the exterior will be empty.
            </documentation>
        </annotation>
    </element>
    <!-- ================================================== -->
    <complexType name="AbstractRingPropertyType">
        <annotation>
            <documentation>Encapsulates a ring to represent the surface boundary property of a surface.</documentation>
        </annotation>
        <sequence>
            <element ref="gml:_Ring"/>
        </sequence>
    </complexType>
    <!-- ================================================== -->
    <element name="_Ring" type="gml:AbstractRingType" abstract="true" substitutionGroup="gml:_Geometry">
        <annotation>
            <documentation>The "_Ring" element is the abstract head of the substituition group for all closed boundaries of a surface
            patch.
            </documentation>
        </annotation>
    </element>
    <!-- ================================================== -->
    <complexType name="AbstractRingType" abstract="true">
        <annotation>
            <documentation>An abstraction of a ring to support surface boundaries of different complexity.</documentation>
        </annotation>
        <complexContent>
            <extension base="gml:AbstractGeometryType"/>
        </complexContent>
    </complexType>
    <!-- ================================================== -->
    <element name="_Geometry" type="gml:AbstractGeometryType" abstract="true" substitutionGroup="gml:_GML">
        <annotation>
            <documentation>The "_Geometry" element is the abstract head of the substituition group for all geometry elements of GML 3. This
            includes pre-defined and user-defined geometry elements. Any geometry element must be a direct or indirect extension/restriction
            of AbstractGeometryType and must be directly or indirectly in the substitution group of "_Geometry".
            </documentation>
            <appinfo>
                <sch:pattern name="Check SRS tags">
                    <sch:rule context="gml:_Geometry">
                        <sch:extends rule="CRSLabels"/>
                    </sch:rule>
                </sch:pattern>
            </appinfo>
        </annotation>
    </element>
    <!-- ================================================== -->
    <element name="_GML" type="gml:AbstractGMLType" abstract="true" substitutionGroup="gml:_Object">
        <annotation>
            <documentation>Global element which acts as the head of a substitution group that may include any element which is a GML feature,
            object, geometry or complex value
            </documentation>
        </annotation>
    </element>
    <!-- ================================================== -->
    <element name="_Object" abstract="true">
        <annotation>
            <documentation>This abstract element is the head of a substitutionGroup hierararchy which may contain either simpleContent or
            complexContent elements.  It is used to assert the model position of "class" elements declared in other GML schemas.
            </documentation>
        </annotation>
</element>
<!-- ================================================== -->
<element name="interior" type="gml:AbstractRingPropertyType">
    <annotation>
        <documentation>A boundary of a surface consists of a number of rings. The "interior" rings seperate the surface / surface patch from
        the area enclosed by the rings.
        </documentation>
    </annotation>
</element>
<!-- ================================================== -->
<element name="_Surface" type="gml:AbstractSurfaceType" abstract="true" substitutionGroup="gml:_GeometricPrimitive">
    <annotation>
        <documentation>The "_Surface" element is the abstract head of the substituition group for all (continuous) surface elements.</documentation>
    </annotation>
</element>
<!-- ================================================== -->
<element name="_GeometricPrimitive" type="gml:AbstractGeometricPrimitiveType" abstract="true" substitutionGroup="gml:_Geometry">
    <annotation>
        <documentation>The "_GeometricPrimitive" element is the abstract head of the substituition group for all (pre- and user-defined)
        geometric primitives.</documentation>
    </annotation>
</element>
<!-- ================================================== -->
<element name="Point" type="gml:PointType" substitutionGroup="gml:_GeometricPrimitive"/>
<!-- ================================================== -->
<complexType name="PointType">
    <annotation>
        <documentation>A Point is defined by a single coordinate tuple.</documentation>
    </annotation>
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType">
            <sequence>
```

```xml
                <choice>
                    <annotation>
                        <documentation>GML supports two different ways to specify the direct positon of a point. 1. The "pos" element is of
                        type DirectPositionType.
                        </documentation>
                    </annotation>
                    <element ref="gml:pos"/>
                    <element ref="gml:coordinates">
                        <annotation>
                            <documentation>Deprecated with GML version 3.1.0 for coordinates with ordinate values that are numbers. Use "pos"
                            instead. The "coordinates" element shall only be used for coordinates with ordinates that require a string
                            representation, e.g. DMS representations.
                            </documentation>
                        </annotation>
                    </element>
                    <element ref="gml:coord">
                        <annotation>
                            <documentation>Deprecated with GML version 3.0. Use "pos" instead. The "coord" element is included for
                            backwards compatibility with GML 2.
                            </documentation>
                        </annotation>
                    </element>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ================================================== -->
<element name="pos" type="gml:DirectPositionType">
    <annotation>
        <appinfo>
            <sch:pattern name="Check SRS tags">
                <sch:rule context="gml:pos">
                    <sch:extends rule="CRSLabels"/>
                </sch:rule>
            </sch:pattern>
        </appinfo>
    </annotation>
</element>
<!-- ================================================== -->
<complexType name="DirectPositionType">
    <annotation>
        <documentation>DirectPosition instances hold the coordinates for a position within some coordinate reference system (CRS). Since
        DirectPositions, as data types, will often be included in larger objects (such as geometry elements) that have references to CRS, the
        "srsName" attribute will in general be missing, if this particular DirectPosition is included in a larger element with such a reference
        to a CRS. In this case, the CRS is implicitly assumed to take on the value of the containing object's CRS.
        </documentation>
    </annotation>
    <simpleContent>
        <extension base="gml:doubleList">
            <attributeGroup ref="gml:SRSReferenceGroup"/>
        </extension>
    </simpleContent>
</complexType>
<!-- ================================================== -->
<simpleType name="doubleList">
    <annotation>
        <documentation>XML List based on XML Schema double type. An element of this type contains a space-separated list of double
        values
        </documentation>
    </annotation>
    <list itemType="double"/>
</simpleType>
<!-- ================================================== -->
<!-- ================================================== -->
<element name="coordinates" type="gml:CoordinatesType">
    <annotation>
        <documentation>Deprecated with GML version 3.1.0.</documentation>
    </annotation>
</element>
<!-- ================================================== -->
<complexType name="CoordinatesType">
    <annotation>
        <documentation>Tables or arrays of tuples.
    May be used for text-encoding of values from a table.
    Actually just a string, but allows the user to indicate which characters are used as separators.
    The value of the 'cs' attribute is the separator for coordinate values,
    and the value of the 'ts' attribute gives the tuple separator (a single space by default);
    the default values may be changed to reflect local usage.
    Defaults to CSV within a tuple, space between tuples.
    However, any string content will be schema-valid.  </documentation>
    </annotation>
    <simpleContent>
        <extension base="string">
            <attribute name="decimal" type="string" default="."/>
            <attribute name="cs" type="string" default=","/>
            <attribute name="ts" type="string" default=" "/>
        </extension>
    </simpleContent>
</complexType>
<!-- ================================================== -->
<element name="coord" type="gml:CoordType">
    <annotation>
        <documentation>Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use the "pos" element instead.</documentation>
    </annotation>
</element>
<!-- ================================================== -->
<complexType name="CoordType">
    <annotation>
        <documentation>Represents a coordinate tuple in one, two, or three dimensions. Deprecated with GML 3.0 and replaced by
        DirectPositionType.</documentation>
```

```xml
    </annotation>
    <sequence>
        <element name="X" type="decimal"/>
        <element name="Y" type="decimal" minOccurs="0"/>
        <element name="Z" type="decimal" minOccurs="0"/>
    </sequence>
</complexType>
<!-- ================================================= -->
<element name="LineString" type="gml:LineStringType" substitutionGroup="gml:_Curve"/>
<!-- ================================================= -->
<complexType name="LineStringType">
    <annotation>
        <documentation>A LineString is a special curve that consists of a single segment with linear interpolation. It is defined by two
        or more coordinate tuples, with linear interpolation between them. It is backwards compatible with the LineString of GML 2,
        GM_LineString of ISO 19107 is implemented by LineStringSegment.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="gml:AbstractCurveType">
            <sequence>
                <choice>
                    <annotation>
                        <documentation>GML supports two different ways to specify the control points of a line string. 1. A sequence of "pos"
                        (DirectPositionType) or "pointProperty" (PointPropertyType) elements. "pos" elements are control points that are only
                        part of this curve, "pointProperty" elements contain a point that may be referenced from other geometry elements or
                        reference another point defined outside of this curve (reuse of existing points). 2. The "posList" element allows for a
                        compact way to specifiy the coordinates of the control points, if all control points are in the same coordinate
                        reference systems and belong to this curve only. The number of direct positions in the list must be at least two.
                        </documentation>
                    </annotation>
                    <choice minOccurs="2" maxOccurs="unbounded">
                        <element ref="gml:pos"/>
                        <element ref="gml:pointProperty"/>
                        <element ref="gml:pointRep">
                            <annotation>
                                <documentation>Deprecated with GML version 3.1.0. Use "pointProperty" instead. Included for backwards
                                compatibility with GML 3.0.0.
                                </documentation>
                            </annotation>
                        </element>
                        <element ref="gml:coord">
                            <annotation>
                                <documentation>Deprecated with GML version 3.0. Use "pos" instead. The "coord" element is included for backwards
                                compatibility with GML 2.
                                </documentation>
                            </annotation>
                        </element>
                    </choice>
                    <element ref="gml:posList"/>
                    <element ref="gml:coordinates">
                        <annotation>
                            <documentation>Deprecated with GML version 3.1.0. Use "posList" instead.</documentation>
                        </annotation>
                    </element>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ================================================= -->
<complexType name="AbstractCurveType" abstract="true">
    <annotation>
        <documentation>An abstraction of a curve to support the different levels of complexity. The curve can always be viewed as a geometric
        primitive, i.e. is continuous.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType"/>
    </complexContent>
</complexType>
<!-- ================================================= -->
<element name="pointProperty" type="gml:PointPropertyType">
    <annotation>
        <appinfo>
            <sch:pattern name="Check either href or content not both">
                <sch:rule context="gml:pointProperty">
                    <sch:extends rule="hrefOrContent"/>
                </sch:rule>
            </sch:pattern>
        </appinfo>
        <documentation>This property element either references a point via the XLink-attributes or contains the point element. pointProperty
        is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that
        is substitutable for Point.
        </documentation>
    </annotation>
</element>
<!-- ================================================= -->
<complexType name="PointPropertyType">
    <annotation>
        <documentation>A property that has a point as its value domain can either be an appropriate geometry element encapsulated in an
        element of this type or an XLink reference to a remote geometry element (where remote includes geometry elements located
        elsewhere in the same document). Either the reference or the contained element must be given, but neither both nor none.
        </documentation>
    </annotation>
    <sequence minOccurs="0">
        <element ref="gml:Point"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup">
        <annotation>
```

```xml
				<annotation>
					<documentation>This attribute group includes the XLink attributes (see xlinks.xsd). XLink is used in GML to reference remote
					resources (including those elsewhere in the same document). A simple link element can be constructed by including a specific
					set of XLink attributes. The XML Linking Language (XLink) is currently a Proposed Recommendation of the World Wide Web Consortium.
					XLink allows elements to be inserted into XML documents so as to create sophisticated links between resources; such links can be
					used to reference remote properties. A simple link element can be used to implement pointer functionality, and this functionality
					has been built into various GML 3 elements by including the gml:AssociationAttributeGroup.
					</documentation>
				</annotation>
			</attributeGroup>
		</complexType>
		<!-- ================================================== -->
		<element name="pointRep" type="gml:PointPropertyType">
			<annotation>
				<documentation>Deprecated with GML version 3.1.0. Use "pointProperty" instead. Included for backwards compatibility
				with GML 3.0.0.
				</documentation>
			</annotation>
		</element>
		<!-- ================================================== -->
		<element name="posList" type="gml:DirectPositionListType">
			<annotation>
				<appinfo>
					<sch:pattern name="Check SRS tags">
						<sch:rule context="gml:posList">
							<sch:extends rule="CRSLabels"/>
						</sch:rule>
					</sch:pattern>
				</appinfo>
				<appinfo>
					<sch:pattern name="Check Dimension">
						<sch:rule context="gml:posList">
							<sch:extends rule="Count"/>
						</sch:rule>
					</sch:pattern>
				</appinfo>
			</annotation>
		</element>
		<!-- ================================================== -->
		<complexType name="DirectPositionListType">
			<annotation>
				<documentation>DirectPositionList instances hold the coordinates for a sequence of direct positions within the same coordinate
				reference system (CRS).
				</documentation>
			</annotation>
			<simpleContent>
				<extension base="gml:doubleList">
					<attributeGroup ref="gml:SRSReferenceGroup"/>
					<attribute name="count" type="positiveInteger" use="optional">
						<annotation>
							<documentation>"count" allows to specify the number of direct positions in the list. If the attribute count is present
							then the attribute srsDimension shall be present, too.
							</documentation>
						</annotation>
					</attribute>
				</extension>
			</simpleContent>
		</complexType>
		<!-- ================================================== -->
		<element name="_Curve" type="gml:AbstractCurveType" abstract="true" substitutionGroup="gml:_GeometricPrimitive">
			<annotation>
				<documentation>The "_Curve" element is the abstract head of the substituition group for all (continuous) curve elements.</documentation>
			</annotation>
		</element>
		<!-- ================================================== -->
		<element name="LinearRing" type="gml:LinearRingType" substitutionGroup="gml:_Ring"/>
		<!-- ================================================== -->
		<complexType name="LinearRingType">
			<annotation>
				<documentation>A LinearRing is defined by four or more coordinate tuples, with linear interpolation between them; the first and last
				coordinates must be coincident.
				</documentation>
			</annotation>
			<complexContent>
				<extension base="gml:AbstractRingType">
					<sequence>
						<choice>
							<annotation>
								<documentation>GML supports two different ways to specify the control points of a linear ring.
								1. A sequence of "pos" (DirectPositionType) or "pointProperty" (PointPropertyType) elements. "pos" elements are
								 control points that are only part of this ring, "pointProperty" elements contain a point that may be referenced
								 from other geometry elements or reference another point defined outside of this ring (reuse of existing points).
								2. The "posList" element allows for a compact way to specifiy the coordinates of the control points, if all
								control points are in the same coordinate reference systems and belong to this ring only. The number of direct
								positions in the list must be at least four.
								</documentation>
							</annotation>
							<choice minOccurs="4" maxOccurs="unbounded">
								<element ref="gml:pos"/>
								<element ref="gml:pointProperty"/>
								<element ref="gml:pointRep">
									<annotation>
										<documentation>Deprecated with GML version 3.1.0. Use "pointProperty" instead. Included for backwards
										compatibility with GML 3.0.0.
										</documentation>
									</annotation>
								</element>
							</choice>
							<element ref="gml:posList"/>
							<element ref="gml:coordinates">
								<annotation>
```

```xml
                            <documentation>Deprecated with GML version 3.1.0. Use "posList" instead.</documentation>
                        </annotation>
                    </element>
                    <element ref="gml:coord" minOccurs="4" maxOccurs="unbounded">
                        <annotation>
                            <documentation>Deprecated with GML version 3.0 and included for backwards compatibility with GML 2.
                            Use "pos" elements instead.
                            </documentation>
                        </annotation>
                    </element>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ================================================= -->
<element name="surfaceProperty" type="gml:SurfacePropertyType">
    <annotation>
        <appinfo>
            <sch:pattern name="Check either href or content not both">
                <sch:rule context="gml:surfaceProperty">
                    <sch:extends rule="hrefOrContent"/>
                </sch:rule>
            </sch:pattern>
        </appinfo>
        <documentation>This property element either references a surface via the XLink-attributes or contains the surface element.
        surfaceProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with
        a value that is substitutable for _Surface.
        </documentation>
    </annotation>
</element>
<!-- ================================================= -->
<complexType name="SurfacePropertyType">
    <annotation>
        <documentation>A property that has a surface as its value domain can either be an appropriate geometry element encapsulated in an
        element of this type or an XLink reference to a remote geometry element (where remote includes geometry elements located elsewhere
        in the same document). Either the reference or the contained element must be given, but neither both nor none.
        </documentation>
    </annotation>
    <sequence minOccurs="0">
        <element ref="gml:_Surface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup">
        <annotation>
            <documentation>This attribute group includes the XLink attributes (see xlinks.xsd). XLink is used in GML to reference remote
            resources (including those elsewhere in the same document). A simple link element can be constructed by including a specific set
            of XLink attributes. The XML Linking Language (XLink) is currently a Proposed Recommendation of the World Wide Web Consortium.
            XLink allows elements to be inserted into XML documents so as to create sophisticated links between resources; such links can be
            used to reference remote properties. A simple link element can be used to implement pointer functionality, and this functionality
            has been built into various GML 3 elements by including the gml:AssociationAttributeGroup.
            </documentation>
        </annotation>
    </attributeGroup>
</complexType>
<!-- ================================================= -->
<!-- ================================================= -->
<element name="_Feature" type="gml:AbstractFeatureType" abstract="true" substitutionGroup="gml:_GML"/>
<!-- ================================================= -->
<complexType name="AbstractFeatureType" abstract="true">
    <annotation>
        <documentation>An abstract feature provides a set of common properties, including id, metaDataProperty, name and description inherited
        from AbstractGMLType, plus boundedBy.    A concrete feature type must derive from this type and specify additional  properties in an
        application schema. A feature must possess an identifying attribute ('id' - 'fid' has been deprecated).
        </documentation>
    </annotation>
    <complexContent>
        <extension base="gml:AbstractGMLType">
            <sequence>
                <element ref="gml:boundedBy" minOccurs="0"/>
                <element ref="gml:location" minOccurs="0">
                    <annotation>
                        <appinfo>deprecated</appinfo>
                        <documentation>deprecated in GML version 3.1</documentation>
                    </annotation>
                </element>
                <!-- additional properties must be specified in an application schema -->
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ================================================= -->
<element name="boundedBy" type="gml:BoundingShapeType"/>
<!-- ================================================= -->
<complexType name="BoundingShapeType">
    <annotation>
        <documentation>Bounding shape.</documentation>
    </annotation>
    <sequence>
        <choice>
            <element ref="gml:Envelope"/>
            <element ref="gml:Null"/>
        </choice>
    </sequence>
</complexType>
<!-- ================================================= -->
<element name="Envelope" type="gml:EnvelopeType"/>
<!-- ================================================= -->
<complexType name="EnvelopeType">
    <annotation>
```

```xml
            <documentation>Envelope defines an extent using a pair of positions defining opposite corners in arbitrary dimensions. The first direct
            position is the "lower corner" (a coordinate position consisting of all the minimal ordinates for each dimension for all points within
            the envelope), the second one the "upper corner" (a coordinate position consisting of all the maximal ordinates for each dimension for
            all points within the envelope).
            </documentation>
        </annotation>
        <choice>
            <sequence>
                <element name="lowerCorner" type="gml:DirectPositionType"/>
                <element name="upperCorner" type="gml:DirectPositionType"/>
            </sequence>
            <element ref="gml:coord" minOccurs="2" maxOccurs="2">
                <annotation>
                    <appinfo>deprecated</appinfo>
                    <documentation>deprecated with GML version 3.0</documentation>
                </annotation>
            </element>
            <element ref="gml:pos" minOccurs="2" maxOccurs="2">
                <annotation>
                    <appinfo>deprecated</appinfo>
                    <documentation>Deprecated with GML version 3.1. Use the explicit properties "lowerCorner" and "upperCorner"
                    instead.
                    </documentation>
                </annotation>
            </element>
            <element ref="gml:coordinates">
                <annotation>
                    <documentation>Deprecated with GML version 3.1.0. Use the explicit properties "lowerCorner" and "upperCorner"
                    instead.
                    </documentation>
                </annotation>
            </element>
        </choice>
        <attributeGroup ref="gml:SRSReferenceGroup"/>
    </complexType>
</schema>
```

Figure 3. 4 XML Schema for Encoding Survey Data

3.3.4 Writing the Web Application

The programming languages and style sheet language used is as shown in Figure 3.4. The code was uploaded to a private account in GitHub (a website for developers and programmers) to automatically generate the composition as shown in Figure 3.5.

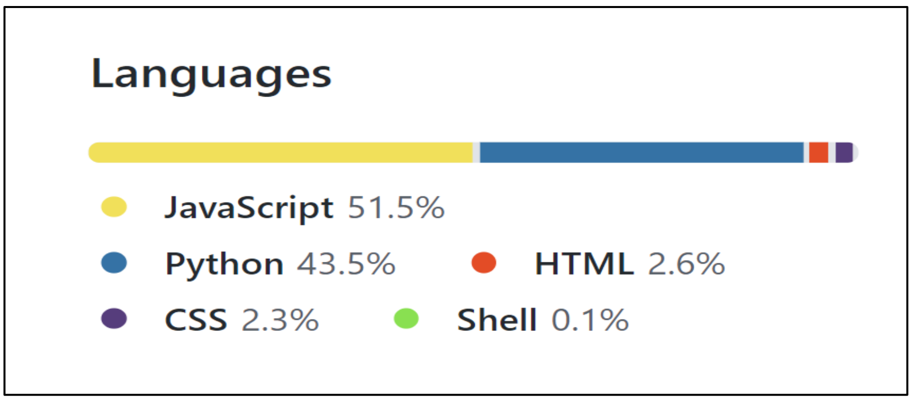


Figure 3. 5 Programming Languages Used

Python was used for backend operations, JavaScript for manipulating data and the dataset uploading form and HTML bootstrap helped in the design of the User Interface (UI). This study embraced bootstrap because it is an enormous collection of handy, reusable bits of code written in JavaScript, HTML and CSS. As a frontend development framework, it enabled a quick buildup of fully responsive functions for the application.

CHAPTER 4: RESULTS AND ANALYSIS

4.1 Key Data Elements of the Fixed Boundary Survey Plan

Key data elements of the Fixed Boundary Cadastral plan that need encoding in the InfraGML survey data interchange standard are as seen in Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.4 and Figure 4.5.

(i) Survey Plan Number also known as the F/R number.

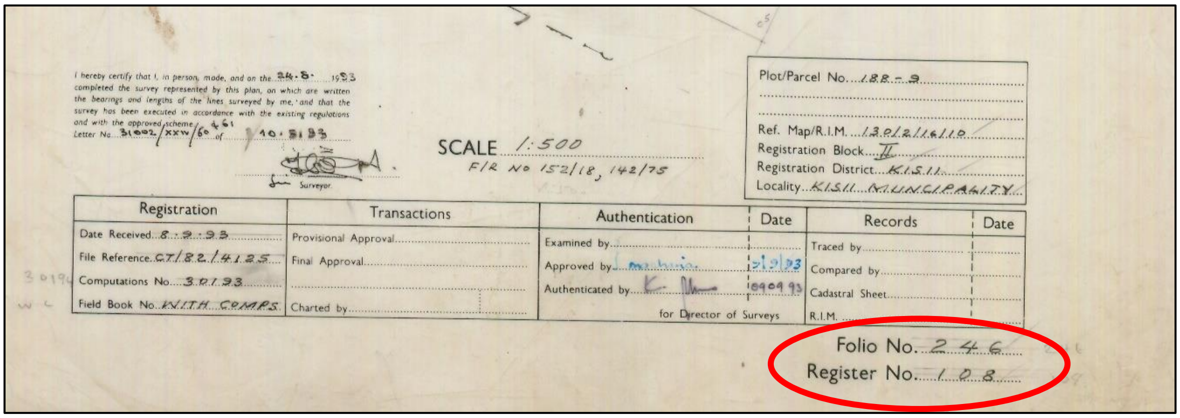


Figure 4. 1 The Folio Register Number of the Survey Plan

(ii) Parcel Number

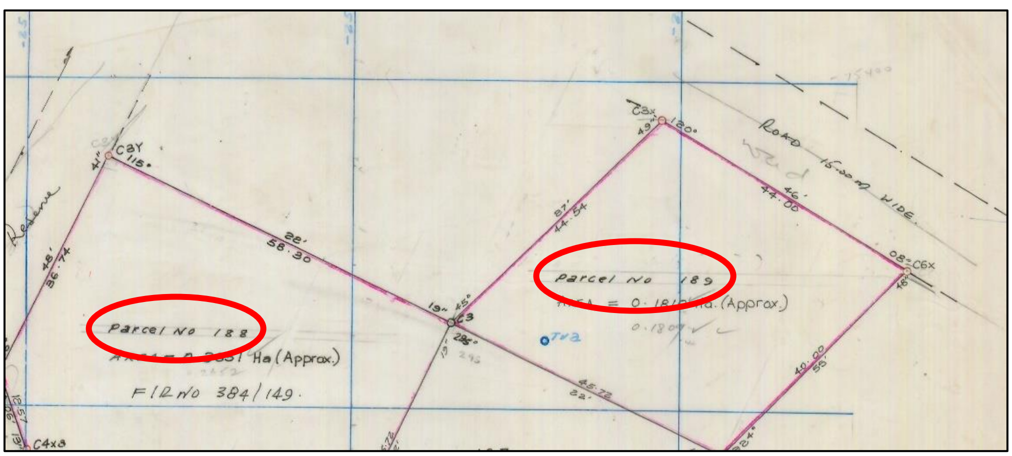
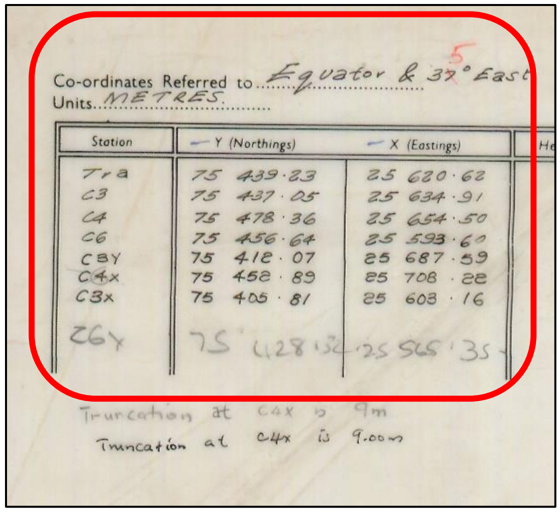


Figure 4. 2 Parcel Numbers for Two Parcels in the Cadastral plan

(iii) Parcel Coordinates



Co-ordinates Referred to Equator & 37° East

Units METRES

Station	Y (Northings)	X (Eastings)
Tra	75 439·23	25 620·62
C3	75 437·05	25 634·91
C4	75 478·36	25 654·50
C6	75 456·64	25 593·60
CBY	75 412·07	25 687·59
C4x	75 452·89	25 708·28
C3x	75 405·81	25 603·16
C6x	75 428·52	25 565·35

Truncation at C4x is 9m

Truncation at C4x is 9.00m

Figure 4. 3 Control Points' Coordinates for the Parcels in the Survey Plan

(iv) Road Reserves

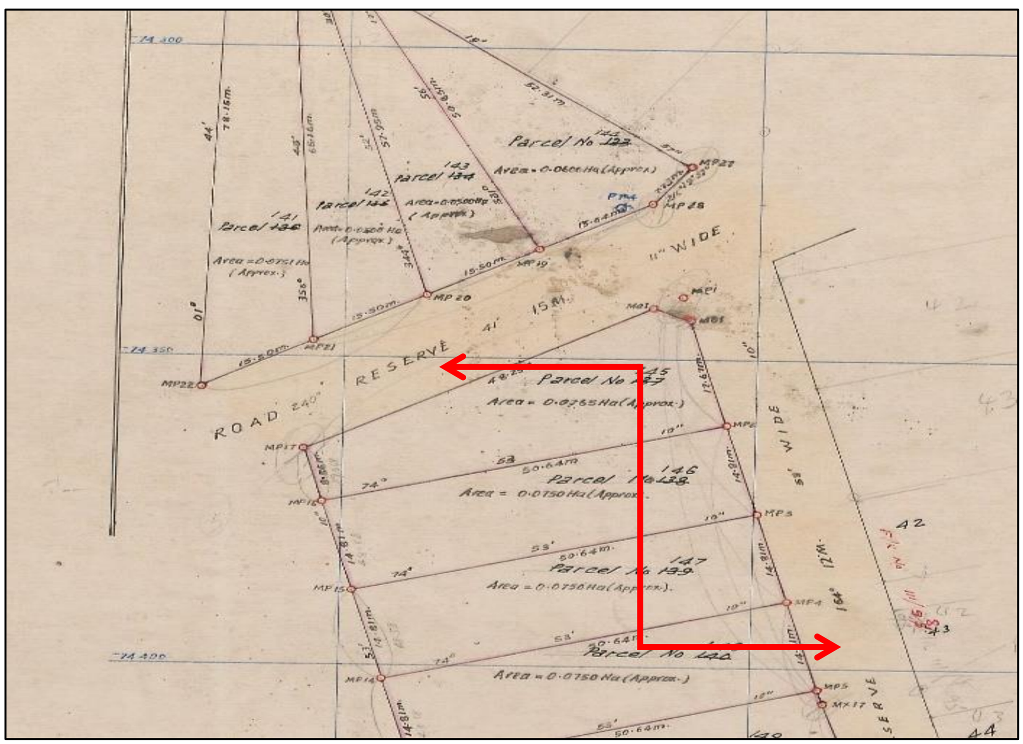


Figure 4. 4 Road Reserve in Folio Register 225/51

(v) Easements

Easements were encoded in the same way as the Road Reserves.

(vi) Abuttals

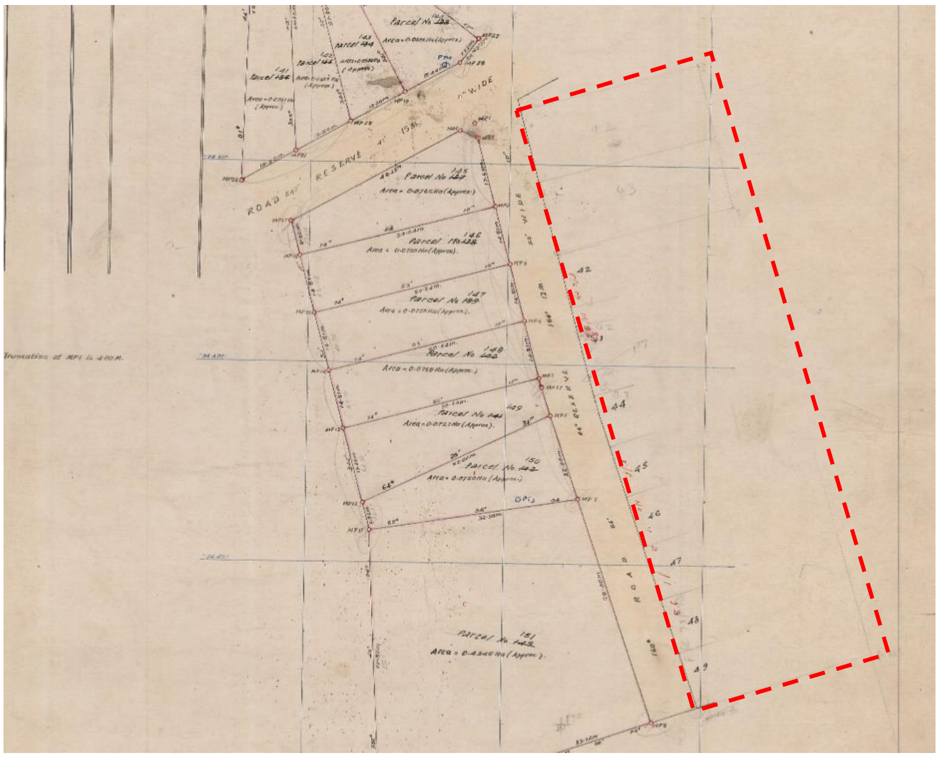


Figure 4. 5 Abuttals in Folio Register 225/51

The encoding of the abuttals was done in the same way as the respective survey plan dataset. It can be done separately or as one dataset.

4.2 The InfraGML Framework for Encoding the Data Elements

The framework is per the alignment of XML elements which are in accordance with the Geography Markup Language XML Schema Definition (GML XSD) specified in InfraGML schemas by OGC LandInfra Standards Working Group (SWG) in OGC - Consortium Model 15-111. The developed InfraGML schema for encoding the data elements will produce results as shown in Figure 4.5.

The encoding in InfraGML survey data interchange standard as generated by the developed InfraGML encoding python program using the schema is as follows:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<LandInfraDataset xmlns="http://www.opengis.net/infragml/core/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xAL="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:lilf="http://www.opengis.net/infragml/landfeature/1.0"
xmlns:gmllr="http://www.opengis.net/gml/3.3/lr"
xmlns:tin="http://www.opengis.net/gml/3.3/tin"
xmlns:li="http://www.opengis.net/infragml/core/1.0"
xmlns:lif="http://www.opengis.net/infragml/facility/1.0"
xsi:schemaLocation="http://www.opengis.net/infragml/landfeature/1.0
http://schemas.opengis.net/infragml/part1/1.0/land-feature.xsd
http://www.opengis.net/infragml/core/1.0 http://schemas.opengis.net/infragml/part0/1.0/core.xsd
http://www.opengis.net/infragml/facility/1.0 http://schemas.opengis.net/infragml/part2/1.0/facility.xsd"
gml:id="GML_a1531350-7af3-494e-af4b-91154a26ce2c">

  <datasetID>
    <ID>
      <identifier>GML_a1531350-7af3-494e-af4b-91154a26ce2c</identifier>
      <scope>OGC LandInfraSWG</scope>
    </ID>
  </datasetID>
  <name>Prince Mwamburi</name>
  <description>Created by me</description>
  <dateTime>2021-07-02T21:21:14</dateTime>
  <datasetVersion>1.0</datasetVersion>

  <infraVersion>1.0</infraVersion>
  <language>English</language>
  <defaultCRS xlink:href="urn:adv:crs:Arc_1960_UTM_zone_36S"/>
  <feature>
    <lif:Facility gml:id="Facility_$3450645">
      <gml:name>Fr_225_51</gml:name>
      <lif:facilityID>
        <lif:ID>
          <identifier>Facility_$3450645</identifier>
          <scope>OGC LandInfra SWG</scope>
        </lif:ID>
      </lif:facilityID>
      <lif:type>Parcel</lif:type>
      <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#existing" xlink:title="existing"/>
      <lif:part xlink:href="#$3450645"/>
    </lif:Facility>
  </feature>
  <feature>
    <lif:Facility gml:id="$3450645">
      <gml:name>Fr_225_51</gml:name>
      <spatialRepresentation>
        <SpatialRepresentation>
          <geometry>
            <gml:Solid>
              <gml:exterior>
                <gml:Shell>
                  <gml:surfaceMember>
                    <gml:Polygon>
                      <gml:exterior>
                        <gml:LinearRing>
                          <gml:pos> 696800.32 9925720.14 3.113</gml:pos>
                          <gml:pos> 696768.18 9925708.97 3.113</gml:pos>
                          <gml:pos> 696768.18 9925708.97 0</gml:pos>
                          <gml:pos> 696800.32 9925720.14 0</gml:pos>
                          <gml:pos> 696800.32 9925720.14 3.113</gml:pos>
                        </gml:LinearRing>
                      </gml:exterior>
                    </gml:Polygon>
                  </gml:surfaceMember>
                  <gml:surfaceMember>
                    <gml:Polygon>
                      <gml:exterior>
                        <gml:LinearRing>
                          <gml:pos> 696768.18 9925708.97 3.113</gml:pos>
                          <gml:pos> 696760.55 9925691.38 3.113</gml:pos>
                          <gml:pos> 696760.55 9925691.38 0</gml:pos>
                          <gml:pos> 696768.18 9925708.97 0</gml:pos>
                          <gml:pos> 696768.18 9925708.97 3.113</gml:pos>
                        </gml:LinearRing>
                      </gml:exterior>
                    </gml:Polygon>
                  </gml:surfaceMember>
                  <gml:surfaceMember>
                    <gml:Polygon>
                      <gml:exterior>
                        <gml:LinearRing>
                          <gml:pos> 696760.55 9925691.38 3.113</gml:pos>
                          <gml:pos> 696732.90 9925689.53 3.113</gml:pos>
                          <gml:pos> 696732.90 9925689.53 0</gml:pos>
                          <gml:pos> 696760.55 9925691.38 0</gml:pos>
                          <gml:pos> 696760.55 9925691.38 3.113</gml:pos>
                        </gml:LinearRing>
                      </gml:exterior>
                    </gml:Polygon>
```

```xml
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696732.90 9925689.53 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
                        <gml:pos> 696732.90 9925689.53 0</gml:pos>
                        <gml:pos> 696732.90 9925689.53 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696800.32 9925720.14 3.113</gml:pos>
                        <gml:pos> 696800.32 9925720.14 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696800.32 9925720.14 3.113</gml:pos>
                        <gml:pos> 696768.18 9925708.97 3.113</gml:pos>
                        <gml:pos> 696760.55 9925691.38 3.113</gml:pos>
                        <gml:pos> 696732.90 9925689.53 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696800.32 9925720.14 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696800.32 9925720.14 0</gml:pos>
                        <gml:pos> 696768.18 9925708.97 0</gml:pos>
                        <gml:pos> 696760.55 9925691.38 0</gml:pos>
                        <gml:pos> 696732.90 9925689.53 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696800.32 9925720.14 0</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
              </gml:Shell>
            </gml:exterior>
          </gml:Solid>
        </geometry>
      </SpatialRepresentation>
    </spatialRepresentation>
    <lif:facilityPartID>
      <lif:ID>
        <identifier>$3450645</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityPartID>
    <lif:type xlink:href="http://www.opengis.net/infragml/facility/1.0/typeCodelist#Parcel" xlink:title="Parcel"/>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#Existing" xlink:title="Existing"/>
  </lif:Parcel>
</feature>
<feature>
  <lif:Facility gml:id="Facility_$1643259">
    <gml:name>Fr_225_51</gml:name>
    <lif:facilityID>
```

```
      <lif:ID>
        <identifier>Facility_$1643259</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityID>
    <lif:type>Parcel</lif:type>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#existing" xlink:title="existing"/>
    <lif:part xlink:href="#$1643259"/>
  </lif:Facility>
</feature>
<feature>
  <lif:Parcel gml:id="$1643259">
    <gml:name>Fr_225_51</gml:name>
    <spatialRepresentation>
      <SpatialRepresentation>
        <geometry>
          <gml:Solid>
            <gml:exterior>
              <gml:Shell>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 0</gml:pos>
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696773.82 9925796.19 0</gml:pos>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                        <gml:pos> 696728.85 9925767.01 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696781.00 9925775.47 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                        <gml:pos> 696728.85 9925767.01 0</gml:pos>
```

```xml
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696773.82 9925796.19 0</gml:pos>
                        <gml:pos> 696781.00 9925775.47 0</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
              </gml:Shell>
            </gml:exterior>
          </gml:Solid>
        </geometry>
      </SpatialRepresentation>
    </spatialRepresentation>
    <lif:facilityPartID>
      <lif:ID>
        <identifier>$1643259</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityPartID>
    <lif:type xlink:href="http://www.opengis.net/infragml/facility/1.0/typeCodelist#Parcel" xlink:title="Parcel"/>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#Existing" xlink:title="Existing"/>
  </lif:Parcel>
</feature>
<feature>
  <lif:Facility gml:id="Facility_$9989636">
    <gml:name>Fr_225_51</gml:name>
    <lif:facilityID>
      <lif:ID>
        <identifier>Facility_$9989636</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityID>
    <lif:type>Parcel</lif:type>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#existing" xlink:title="existing"/>
    <lif:part xlink:href="#$9989636"/>
  </lif:Facility>
</feature>
<feature>
  <lif:Parcel gml:id="$9989636">
    <gml:name>Fr_225_51</gml:name>
    <spatialRepresentation>
      <SpatialRepresentation>
        <geometry>
          <gml:Solid>
            <gml:exterior>
              <gml:Shell>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696771.52 9925803.25 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 0</gml:pos>
                        <gml:pos> 696771.52 9925803.25 0</gml:pos>
                        <gml:pos> 696771.52 9925803.25 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696773.82 9925796.194 0</gml:pos>
                        <gml:pos> 696773.82 9925796.194 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                        <gml:pos> 696721.91 9925792.29 0</gml:pos>
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
```

```xml
                        <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                        <gml:pos> 696770.84 9925805.52 0</gml:pos>
                        <gml:pos> 696721.91 9925792.29 0</gml:pos>
                        <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                        <gml:pos> 696771.52 9925803.25 3.113</gml:pos>
                        <gml:pos> 696771.52 9925803.25 0</gml:pos>
                        <gml:pos> 696770.84 9925805.52 0</gml:pos>
                        <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696771.52 9925803.25 3.113</gml:pos>
                        <gml:pos> 696773.82 9925796.19 3.113</gml:pos>
                        <gml:pos> 696727.07 9925773.76 3.113</gml:pos>
                        <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                        <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                        <gml:pos> 696771.52 9925803.25 3.113</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
                <gml:surfaceMember>
                  <gml:Polygon>
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pos> 696771.52 9925803.25 0</gml:pos>
                        <gml:pos> 696773.82 9925796.19 0</gml:pos>
                        <gml:pos> 696727.07 9925773.76 0</gml:pos>
                        <gml:pos> 696721.91 9925792.29 0</gml:pos>
                        <gml:pos> 696770.84 9925805.52 0</gml:pos>
                        <gml:pos> 696771.52 9925803.258 0</gml:pos>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </gml:surfaceMember>
              </gml:Shell>
            </gml:exterior>
          </gml:Solid>
        </geometry>
      </SpatialRepresentation>
    </spatialRepresentation>
    <lif:facilityPartID>
      <lif:ID>
        <identifier>$9989636</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityPartID>
    <lif:type xlink:href="http://www.opengis.net/infragml/facility/1.0/typeCodelist#Parcel" xlink:title="Parcel"/>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#Existing" xlink:title="Existing"/>
  </lif:Parcel>
</feature>
<feature>
  <lif:Facility gml:id="Facility_$3450645">
    <gml:name>Fr_225_51</gml:name>
    <lif:facilityID>
      <lif:ID>
        <identifier>Facility_$3450645</identifier>
        <scope>OGC LandInfra SWG</scope>
      </lif:ID>
    </lif:facilityID>
    <lif:type>Parcel</lif:type>
    <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#existing" xlink:title="existing"/>
    <lif:part xlink:href="#$3450645"/>
  </lif:Facility>
</feature>
<feature>
  <lif:Parcel gml:id="$3450645">
    <gml:name>Fr_225_51</gml:name>
    <spatialRepresentation>
      <SpatialRepresentation>
        <geometry>
          <gml:Solid>
            <gml:exterior>
              <gml:Shell>
                <gml:surfaceMember>
                  <gml:Polygon>
```

```xml
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                <gml:pos> 696721.91 9925792.29 0</gml:pos>
                <gml:pos> 696770.84 9925805.52 0</gml:pos>
                <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                <gml:pos> 696717.94 9925806.74 3.113</gml:pos>
                <gml:pos> 696717.94 9925806.74 0</gml:pos>
                <gml:pos> 696721.91 9925792.29 0</gml:pos>
                <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696717.94 9925806.74 3.113</gml:pos>
                <gml:pos> 696766.91 9925819.72 3.113</gml:pos>
                <gml:pos> 696766.91 9925819.72 0</gml:pos>
                <gml:pos> 696717.94 9925806.74 0</gml:pos>
                <gml:pos> 696717.94 9925806.74 3.113</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696766.91 9925819.72 3.113</gml:pos>
                <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                <gml:pos> 696770.84 9925805.52 0</gml:pos>
                <gml:pos> 696766.91 9925819.72 0</gml:pos>
                <gml:pos> 696766.91 9925819.72 3.113</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
                <gml:pos> 696721.91 9925792.29 3.113</gml:pos>
                <gml:pos> 696717.94 9925806.74 3.113</gml:pos>
                <gml:pos> 696766.91 9925819.72 3.113</gml:pos>
                <gml:pos> 696770.84 9925805.52 3.113</gml:pos>
              </gml:LinearRing>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:pos> 696705.62 9925926.23 0</gml:pos>
                <gml:pos> 696708.97 9925861.64 0</gml:pos>
                <gml:pos> 696695.39 9925853.89 0</gml:pos>
                <gml:pos> 696698.10 9925931.78 0</gml:pos>
                <gml:pos> 696705.62 9925926.23 0</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:Shell>
    </gml:exterior>
  </gml:Solid>
  </geometry>
  </SpatialRepresentation>
  </spatialRepresentation>
  <lif:facilityPartID>
    <lif:ID>
      <identifier>$23417853</identifier>
      <scope>OGC LandInfra SWG</scope>
    </lif:ID>
  </lif:facilityPartID>
  <lif:type xlink:href="http://www.opengis.net/infragml/facility/1.0/typeCodelist#Parcel" xlink:title="Parcel"/>
  <lif:status xlink:href="http://www.opengis.net/infragml/facility/1.0/statusCodelist#Existing" xlink:title="Existing"/>
  </lif:Parcel>
  </feature>
</LandInfraDataset>
```


Figure 4. 6 Resultant InfraGML File

4.3 Demonstrating Cadastral Data interchange with QGIS.

Figure 4.6 depicts the User Interface (UI) of the application wich encompasses programs to aid in:

(i) Encoding of key data elements of a cadastral plan as per the developed InfraGML framework for encoding survey data in Kenya.

(ii) Merging the uploaded dataset with additional important survey report.

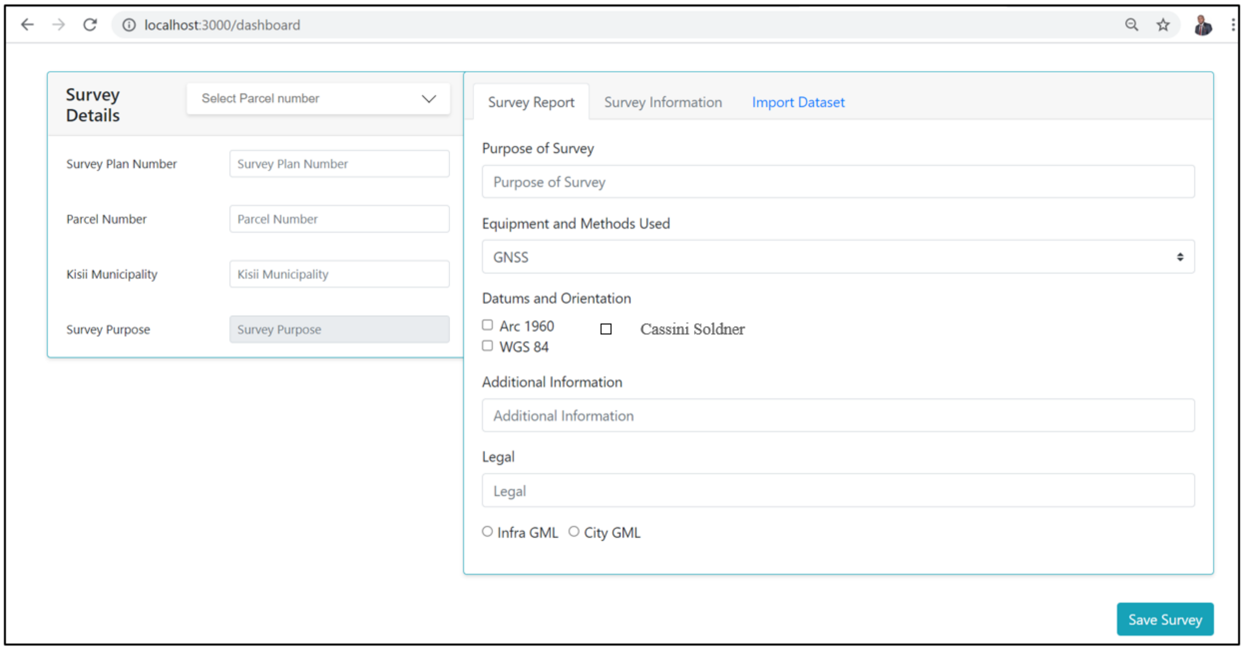


Figure 4. 7 User Interface (UI) of the Application

All that a surveyor needs to do is to upload the dataset and the survey report as seen in the sections of the UI and the application generates the InfraGML file. In the backend the encoding happens as per the GML XSD specified in InfraGML schemas by OGC LandInfra Standards Working Group (SWG) in OGC-Consortium Model 15-111.

4.3.1 Uploading the Survey

(i) Survey Report

The survey report upload and encoding process involves attachment of survey report with the respective dataset. The process is as seen in Figure 4.7.

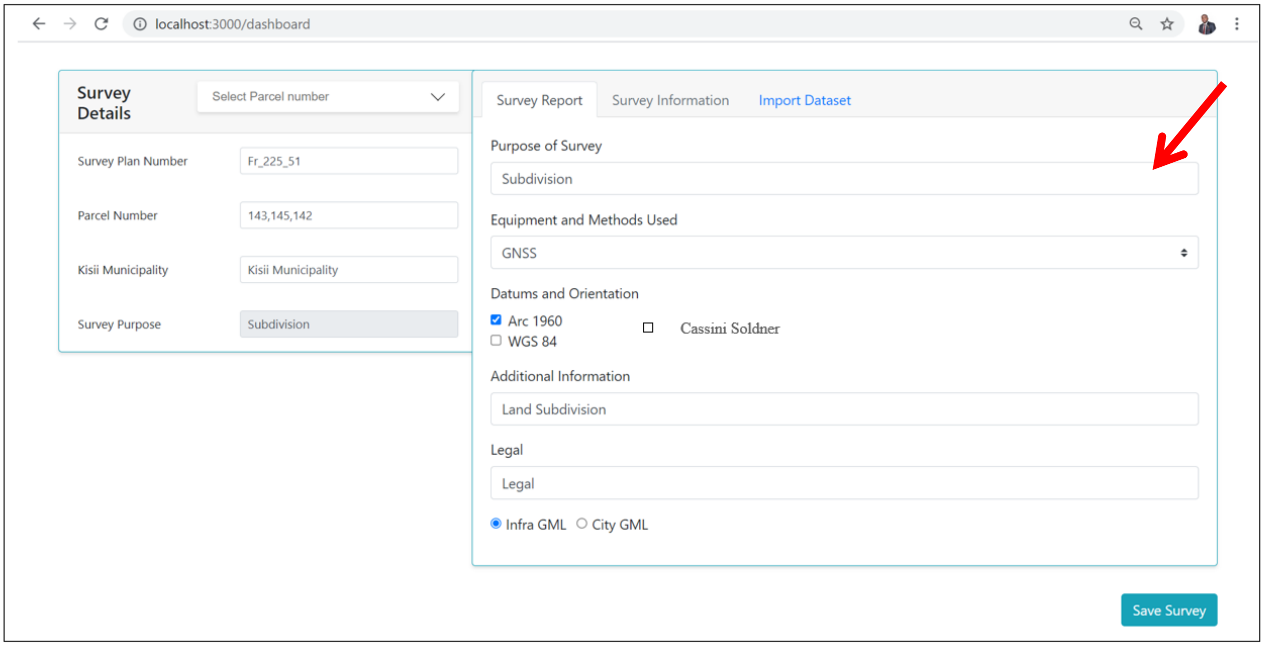


Figure 4. 8 Encoding of Survey Report

(ii) Dataset Input

After inputing the survey report the surveyor navigates to the import dataset section where he/she uploads the layer files. This is as shown in Figure 4.8.

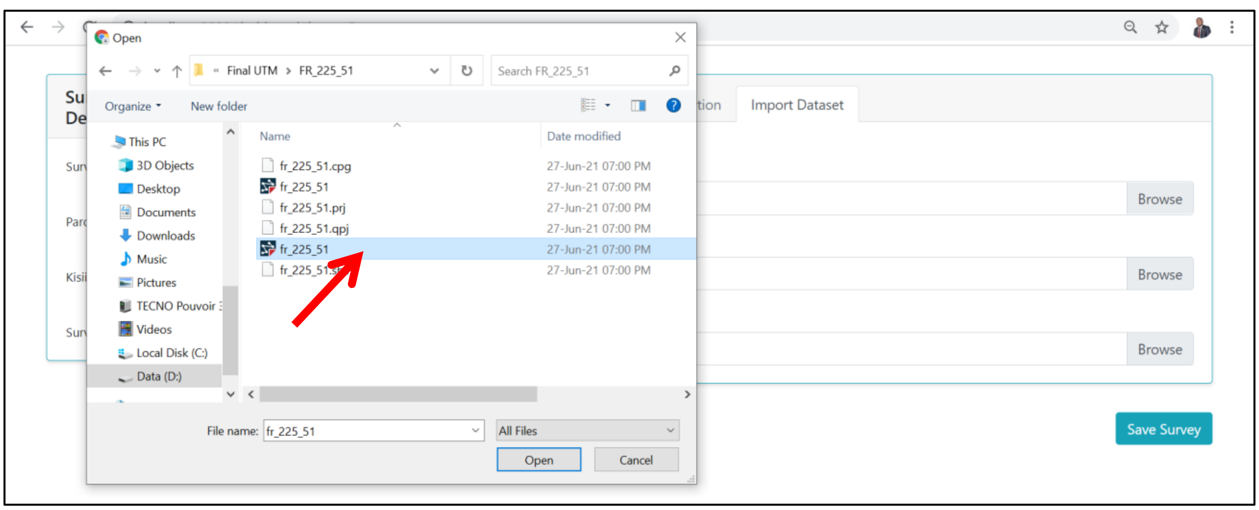


Figure 4. 9 Import Dataset Section

After uploading the required layers the Surveyor clicks on the Save Survey button and the data is encoded in InfraGML and saved in the database. This is depicted in Figure 4.9.

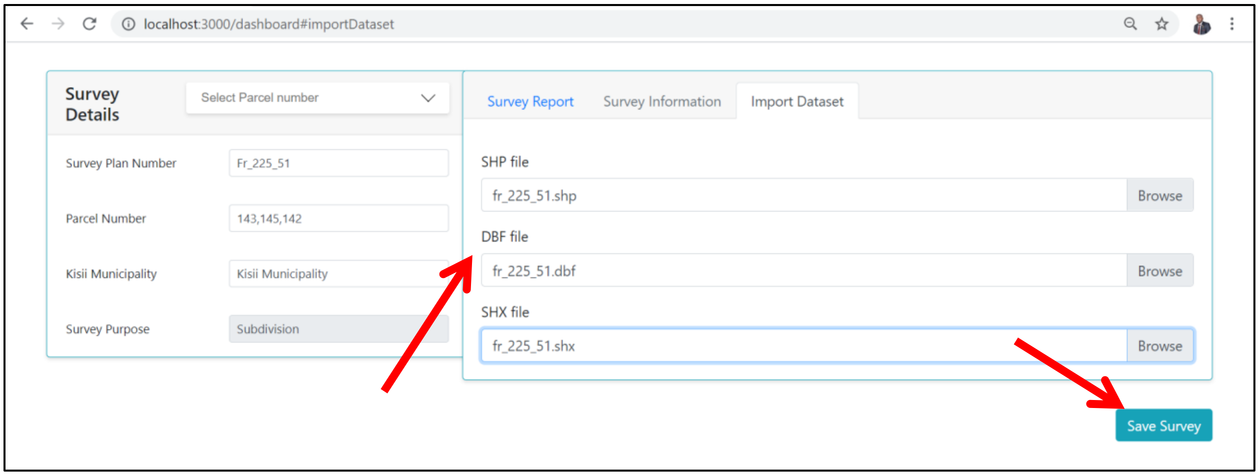


Figure 4. 10 Dataset Input

4.3.2 Data Retrieval

To retrieve data from the database, the surveyor keys in any parcel number that is captured in the survey plan and the survey purpose. The file automatically downloads to their local device storage, as shown in Figure 4.9.

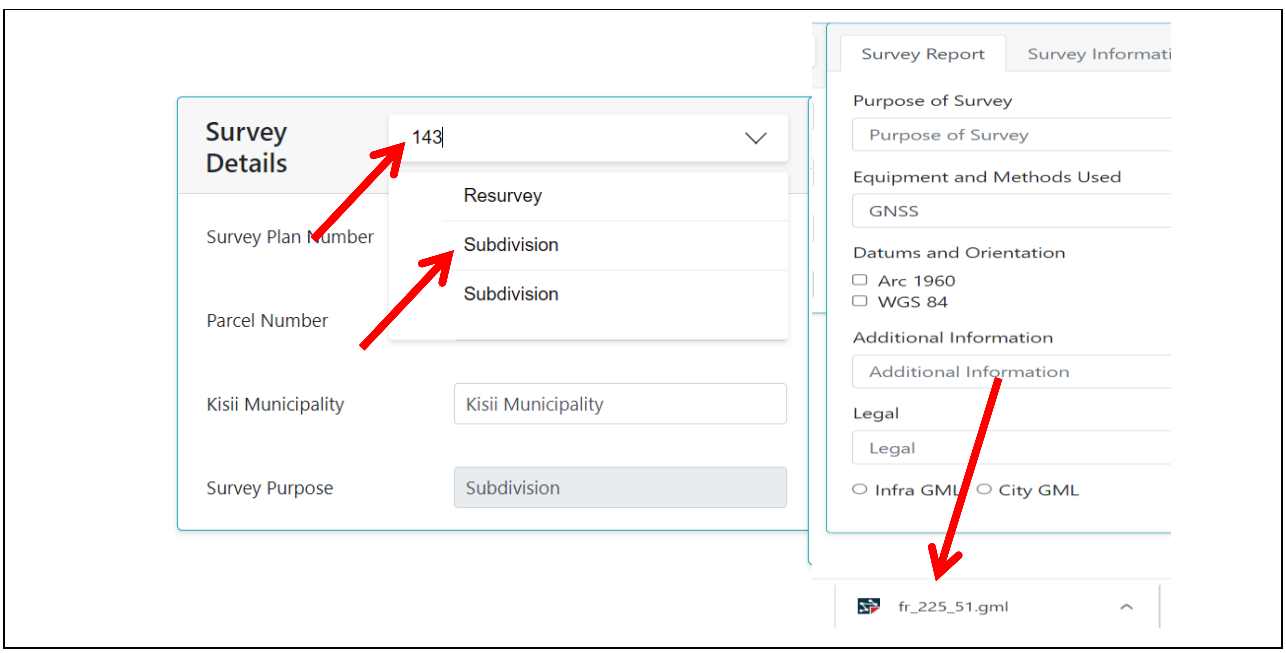


Figure 4. 11 Data Retrieval from the Database

Apart from saving the data in the database the application is also capable of saving data in a local data storage directly after clicking the save survey button on the application's User Interface (UI). The generated file is shown in Figure 4.11.

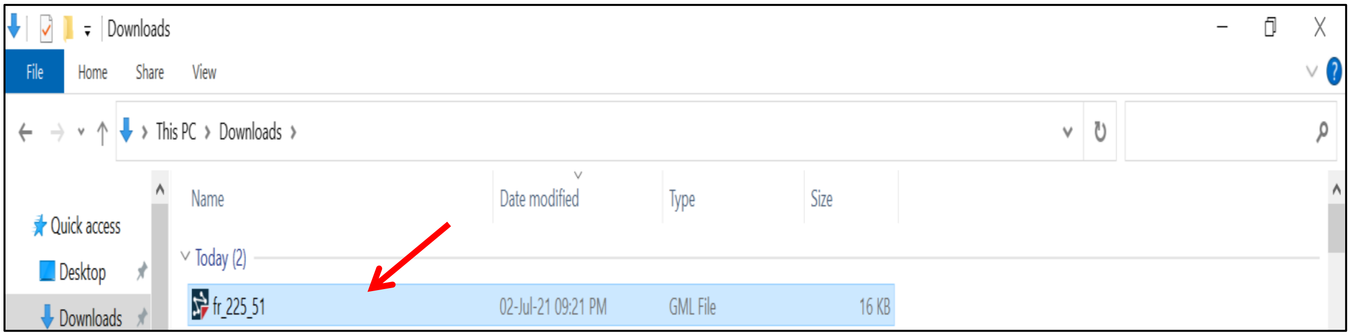


Figure 4. 12 The Resultant InfraGML File

4.3.3 Visualisation

The generated InfraGML file being the survey data for Folio Register No 225_51 was visualised in Quantum GIS. A map was also generated as shown in Figure 4.12.

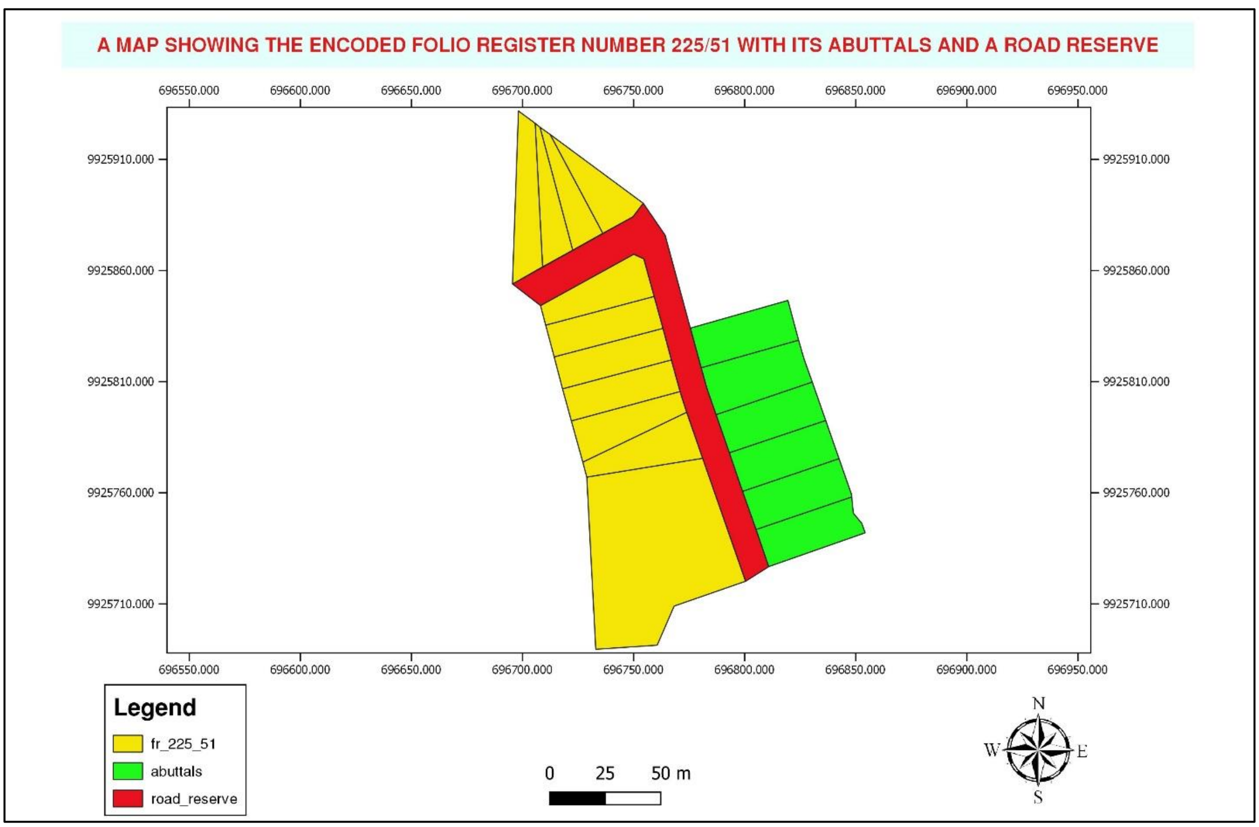


Figure 4. 13 Visualization of F/R No 225/51 Parcels

5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As depicted in the results of the study, new technologies like XML and GML have the ability to make two systems, businesses or multiple organizations transfer geospatial data amongst each other as easy as "Export to InfraGML". Even information search on the internet can be enhanced through XML.

With InfraGML, an XML grammar that specifies geographical features and geospatial datasets, one can import and export data and all items in a geodatabase such as feature datasets, domains, topologies and rules. Validation against the InfraGML schema of XML containing geodatabase data can also be done using automated utilities.

XML helps find a way of interoperability in data interchange with the benefit of independency of any special software. The developed InfraGML standard for encoding and enhancing interchange of survey data has demonstrated that the authoritative nature of the cadastral plan can be maintained even in the absence of a paper document.

5.2 Recommendation

The adoption and integration of this application with the recent developed Land Information Management System named Ardhisasa will provide for standardization in encoding of the cadastral plan data by the surveyors rather than just uploading of shapefiles as it is now in the developed LIMS. Interoperability in storage and transferring of data between systems will also be achieved. With the advancements in technology supporting contemporary geospatial developments, it is important to carry out research in the future on the development of an XML framework outlining procedures based on policies to standardize survey data interchange.

REFERENCES

Admire Nyakudya. (2018). *A new plugin to manage land (survey) parcels in QGIS | Kartoza*. https://kartoza.com/en/blog/a-new-plugin-to-manage-survey-parcels-in-qgis/

Bin Isa, M. N., Hua, T. C., & Binti Abdul Halim, N. Z. (2015). SMARTKADASTER: OBSERVING BEYOND TRADITIONAL CADASTRE CAPABILITIES FOR MALAYSIA. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-2/W4*, 53–55. https://doi.org/10.5194/isprsarchives-XL-2-W4-53-2015

Czarnik, P. (2013). *"Modern" XML applications—XML in electronic data interchange, application integration and databases*. 39.

Esri. (n.d.). *Common COGO workflows—Help | ArcGIS for Desktop*. Retrieved July 2, 2021, from https://desktop.arcgis.com/en/arcmap/10.3/manage-data/creating-new-features/common-cogo-workflows.htm

Foster, E. (2006). *Using web services to exchange information via XML*. 8.

Gröger, G., Kolbe, T. H., Nagel, C., & Häfele, K.-H. (2012). *OGC City Geography Markup Language (CityGML) En- coding Standard*. 344.

https://help.scribesoft.com. (n.d.). *Transferring Data Between Systems Using XML*. Retrieved March 19, 2021, from https://help.scribesoft.com/scribeinsight/en/Subsystems/Insight/create_dts_files/manageconnections/xml/working_with_xml/transferring_data_between_systems_using_xml.htm

ISO. (2008). *ISO 9001:2008(en), Quality management systems—Requirements*. https://www.iso.org/obp/ui/#iso:std:iso:9001:ed-4:v2:en

Ives, Z. G., & Lu, Y. (2000). XML Query Languages in Practice: An Evaluation. In H. Lu & A. Zhou (Eds.), *Web-Age Information Management* (pp. 29–42). Springer. https://doi.org/10.1007/3-540-45151-X_3

Kumar, K., Labetski, A., Ohori, K. A., Ledoux, H., & Stoter, J. (2019). The LandInfra standard and its role in solving the BIM-GIS quagmire. *Open Geospatial Data, Software and Standards*, *4*(1), 5. https://doi.org/10.1186/s40965-019-0065-z

LandMark. (2020). *EuroGeographics_AnnualReview2019_TheNetherlands.pdf*.

Lemmen, C., van Oosterom, P., & Bennett, R. (2015). The Land Administration Domain Model. *Land Use Policy*, *49*, 535–545. https://doi.org/10.1016/j.landusepol.2015.01.014

LINZ and PwC Consulting. (2007). *Landxml-cadastral-survey-exchange-format-v2.0.pdf*.

Mwungu, C. (2020, April 21). *A Subject Domain Model of Kenya's Cadastral System*.

Moran, T. J. (2007). Impact Of XML On Data Interchange: An XML/EDI Model. *Review of Business Information Systems (RBIS)*, *11*(3), 35–42. https://doi.org/10.19030/rbis.v11i3.4414

Open Geospatial Consortium. (2017). *OGC LandInfra / InfraGML | OGC*. https://www.ogc.org/standards/infragml

Siriba, D. N., Voß, W., & Mulaku, G. C. (2011). *The Kenyan Cadastre and Modern Land Administration*. 10.

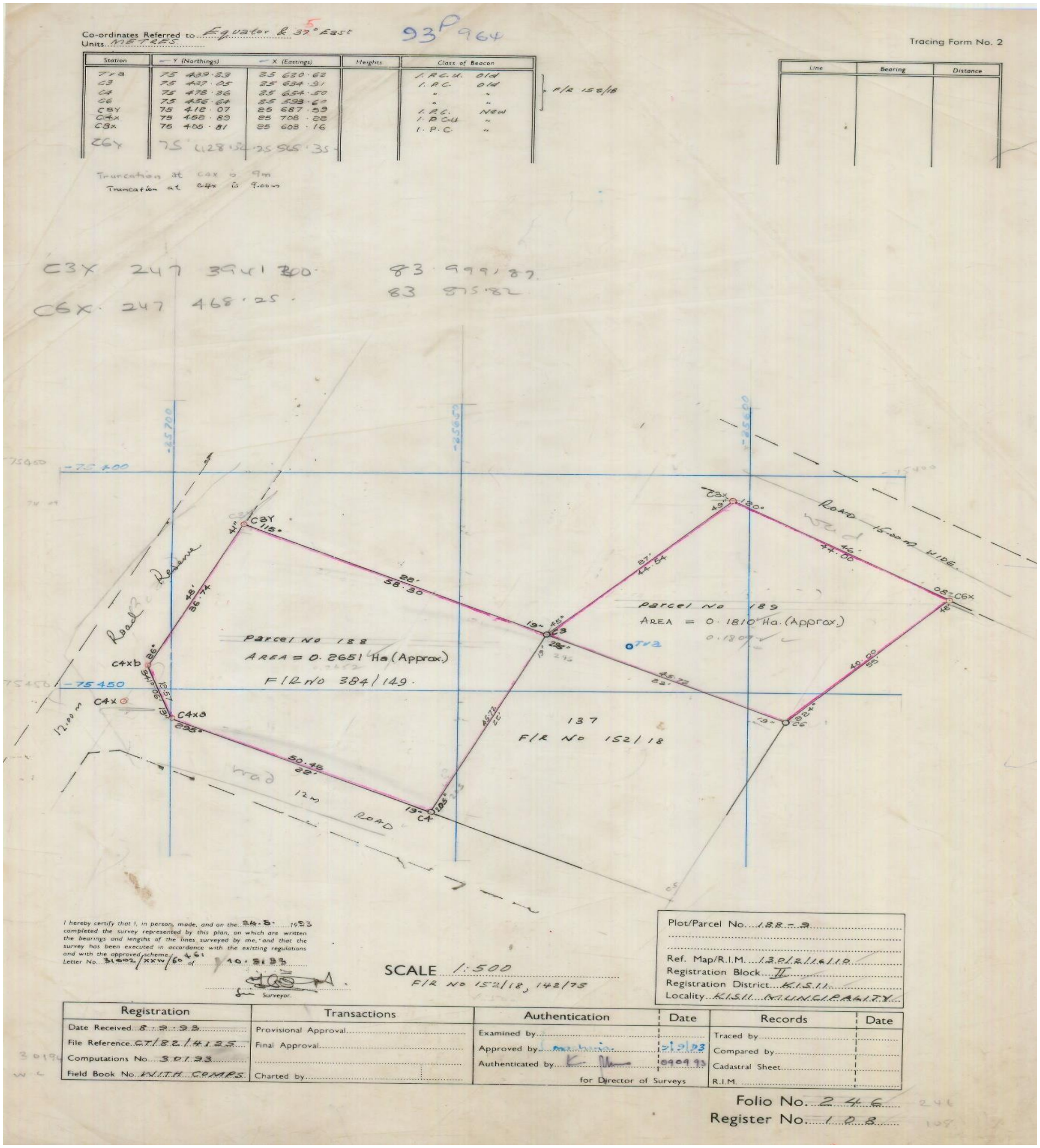
Thompson, R., van Oosterom, P., & Soon, K. (2017). LandXML Encoding of Mixed 2D and 3D Survey Plans with Multi-Level Topology. *ISPRS International Journal of Geo-Information*, *6*(6), 171. https://doi.org/10.3390/ijgi6060171

Wayumba, G. O. (2013). *AN EVALUATION OF THE CADASTRAL SYSTEM IN KENYA AND A STRATEGY FOR ITS MODERNIZATION*. 189.

Webber, D. R. (1998). Introducing xml/edi frameworks. *Electronic Markets*, *8*(1), 38–41.

APPENDIX

A1: Fixed Survey Plan – FR No 246/108



B1: Python Code Linking the Web Application with PostGIS Database

```python
"""
Django settings for api project.

Generated by 'django-admin startproject' using Django 3.2.

For more information on this file, see
https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.2/ref/settings/
"""
import os
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

if os.name == 'nt':
    import platform
    OSGEO4W = r"C:\OSGeo4W"
    if '64' in platform.architecture()[0]:
        OSGEO4W += "64"
    assert os.path.isdir(OSGEO4W), "Directory does not exist: " + OSGEO4W
    os.environ['OSGEO4W_ROOT'] = OSGEO4W
    os.environ['GDAL_DATA'] = OSGEO4W + r"\share\gdal"
    os.environ['PROJ_LIB'] = OSGEO4W + r"\share\proj"
    os.environ['PATH'] = OSGEO4W + r"\bin;" + os.environ['PATH']

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-4q$9b^(a8q3rg7cy2ppvd7t^d3%u)3hxt69gtc&j9k-=($b701'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["ec2-3-7-236-172.ap-south-1.compute.amazonaws.com","127.0.0.1"]


# Application definition

CORS_ALLOW_ALL_ORIGINS = True

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'django.contrib.gis',
    'gis_bank'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'api.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates']
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'api.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

```

```python
DATABASES = {
    'default': {

        'ENGINE': 'django.contrib.gis.db.backends.postgis',

        'NAME': 'gis_db',

        'USER': 'gis_admin',

        'PASSWORD': 'postgis_web',

        'HOST': 'localhost',

        'PORT': '5432',

    }
}


# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```