NOVEC EXpansion Identification System

Final Report

May 8, 2015
Revision E

GEORGE MASON UNIVERSITY

NOVEC
Investing in Our Communities
SM

Austin Orchard
Brian Smith
Tygue Ferrier

NOVEC EXpansion Identification System

Final Report

SYST 699 - Spring 2015

Austin Orchard
Brian Smith
Tygue Ferrier

Version	Revision Status	Date
A	Initial Release	Apr 7, 2015
B	Redlines from Meeting 04/12/15 Meeting	Apr 16, 2015
C	Redlines from NOVEC Review 04/24/15	Apr 27, 2015
D	Redlines from NOVEC 04/28/15	Apr 39, 2015
E	Final Submission	May 8, 2015

Table of Contents [1]

1.0 Executive Summary

The Northern Virginia Electric Cooperative (NOVEC) is a not-for-profit electricity distribution company. NOVEC is one of the largest electric cooperatives in the county, providing service to over 155,000 homes and businesses in the Northern Virginia metropolitan area.

NOVEC regularly conducts load analysis and forecasting to predict future needs and load demands in the near and long terms within its service territory, patches of land in Clark County, Fairfax County, Fauquier County, Loudoun County, Prince Williams County, Stafford County, , and the City of Manassas Park and Town of Clifton. The results of these predictions are one of the bases in deciding where and when to deploy electricity transmission-and-distribution assets and how much electricity to purchase from the PJM market. The spatial load forecasting of NOVEC has not been able to systematically make use of future local development planned and approved by local counties due to the lack of a system and process in capturing, storing, monitoring, and processing local developmental information, such as subdivision approvals and rezoning of certain parcels.

The NOVEC EXpansion Identification System (NEXIS) was designed to fill this gap. NEXIS is an automated data collection tool that collects, stores, processes, and reports large amount of development data across county planning and development agencies to NOVEC data analysts. NEXIS provides three new capabilities to data analysts; data collection, data management, and data reporting of local development within NOVEC service territory systematically and efficiently, rather than in an ad-hoc piecemeal manner. The combination of these new capabilities will allow NOVEC to develop spatial electricity load forecasting models that will systematically use county development data and save time and human resources in keeping up with quickly evolving county development.

2.0 Referenced Documents

1. **IEEE Std. 1362-1998 – IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document**, Institute of Electrical and Electronics Engineers, Inc., New York NY, 1998
2. **Bylaws of the Cooperative**, NOVEC Charter, Dale City VA, 2014
3. **AQS Submit Automation Concept of Operations**, US Environmental Protection Agency Office of Air Quality Planning and Standards Outreach and Information Division National Air Data Group, Research Park NC, 2011
4. **Loudoun Online Land Application Database**, http://www.loudoun.gov/index.aspx?nid=1078, Loudoun County Government, Loudoun County VA, 2015
5. **Loudoun County Available Land Database**, http://biz.loudoun.gov/index.aspx?nid=150, Loudoun County Government: Department of Economic Development, Loudoun County VA, 2015

3.0 Introduction

3.1 Background

The Northern Virginia electric Cooperative (NOVEC) is a not-for-profit electricity distribution company. NOVEC leverages over 30 years of experience in providing electric service to Northern Virginia. NOVEC's service area covers spans Clarke, Fairfax, Fauquier, Loudoun, Prince William, and Stafford counties. This service area includes over 155,000 homes and businesses.

NOVEC does not produce the electricity they sell; rather, they purchase the electricity from power plants and transport it to their customers. NOVEC relies on predictive demand models in order to supply a constant and cost consistent product. The results of these models are used in determining where to place assets and how much energy to purchase from producers.

NOVEC produces monthly forecasting data in order to meet customer demands. These models have two time horizons short term (less than a week) and long term (greater than twenty years). The short and long term models use a time slice of a day and month respectively. Both models serve a unique purpose. The long term forecasting models are used to plan the deployment of future assets and infrastructure. Alternatively, the short-term models are used for purchasing electricity from energy producers and wholesale providers. NOVEC's success is highly dependent on the accuracy of both forecasting models.

Energy usage is dependent on multiple factors including property type, economic status, and weather patterns. Current models use both usage and usage patterns to predict demand. Historical data provides and current short-term models provide accurate predictions. The long-term models have room for improvement. NOVEC has identified at least one information gap, development data.

Figure 1: NOVEC Service Territory

Figure 1 depicts the NOVEC service area in Loudoun County. This area represents a subset of the larger county. NOVEC is only interested in the development within their service area; this results in complications when attempting to collect data concerning ongoing development projects. Local development data is provided by government planning agencies, however, data is produced and reported on a regional level. This data represents average growth statistics for the county. Data at this level of aggregation does not provide the level of detail needed to generate accurate forecasting models.

NOVEC has experimented with the use of development data. Data analysts would manually collect data about known development project. This approach may product errors in data accuracy and completeness. Additionally, collecting and entering data manually is time consuming for personnel whose time could otherwise be leveraged. The current lack of development data imposes the risk of developing models that do not accurately reflect the future demand generated by development projects. This gap produces the need for the development of new capabilities for the NOVEC team.

3.2 Existing System

NOVEC does not currently have a formal system in place for collecting development data. Consideration has previously been given to manually collecting data manually. This consideration has led to the development of two business processes.

The first process utilizes community notifications (e.g. leaflets, town hall meetings, etc.) to trigger data collection. This process is event triggered and provides inconsistent data updates.

Leaflets are often disseminated only within the immediate area that the development impacts. This means that NOVEC may not be notified if they do not have a company building within impact area. This process can lead to multiple projects being over looked because they were not detected. Secondly, this process is exceptionally time consuming. Notification often provide a time and place for a town hall meeting where the development will be discussed. This requires that NOVEC employees attend such meetings to receive details on the development project. The second data collection capability is triggered by construction companies.

The second data collection capability employs the existing communication channels between NOVEC and land development companies. When development companies near the completion of a project they notify NOVEC of the property location and size. This signals to NOVEC that the new property will need to be integrated into the customer pool. NOVEC accordingly develops a service plan for connecting the new property into the distribution network. This process possesses a critical temporal disadvantage. The notification of completion at the end of the project life cycle does not allow for NOVEC to include ongoing projects in forecast models or planning. This results in forecast models that lag the actual development in the service area. NOVEC currently has a third data collection option supported by their current systems.

NOVEC has the ability to automate the entire data collection process through the use of Loudoun County's Land Develop Database. The process would follow a multi-step process: Identify Project Id number, identify project pin, map pin to address, determine if the address is within the NOVEC service area. This process is currently resource prohibitive.

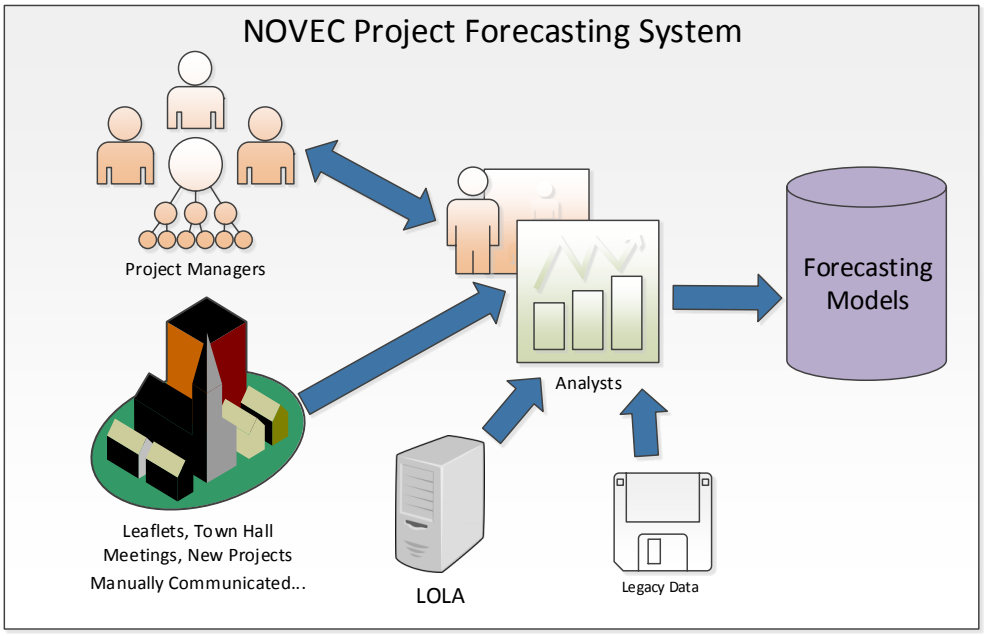


Figure 2: Existing Forecasting System

The process is manual in nature and can be cumbersome. There is very little automation and in a service oriented company that needs to maintain a high level of customer interaction this can lead to inaccurate predictions of project intent and therefore inefficient control for NOVEC's customers. A change in the current system with increased automation is therefore justifiable.

The three existing capabilities offer various levels of situational awareness. However, each process is resource intensive and potentially prone to error. Transcribing hundreds of records by hand may introduce errors that would compromise the fidelity of the data. The inefficient nature of manual data collection has left a gap in NOVEC's ability to analyze their user needs. This gap can be filled through the development of a new system that provides the capability to replace the existing system.

3.3 Statement of Problem

NOVEC intends to make full use of publicly available information on local development to create short term and long term spatial electricity load forecasts and project planning. The spatial load forecasts will advance the current allocation-based load forecasting at the delivery, substation, and circuit level. The current approach does not adequately reflect where development is happening within a county and where energy demands will change. With more localized spatial development data, NOVEC will be able to make more informed prediction of future energy demands and infrastructure needs in the short and long term.

A system needs to be developed to obtain, aggregate data from various county websites periodically, stores relevant data in a database, and generates reports detailing changes to the data and highlighting important existing or new projects. NOVEC will be provided the means to retrieve specific data that is needed to incorporate into their existing forecasting models.

3.4 Scope

This project is to use Loudoun County as a case study and build a prototype database to serve NOVEC's needs stated above. The team will create the database, provide the ability to update the database and provide the ability to create reports focusing on high value projects as determined by NOVEC.

Three planning, zoning, and land development database of Loudoun County are:

1. Planning and Zoning Applications (through Loudoun Online Land Applications Database)
2. Subdivision Applications (through Loudoun Online Land Applications Database)
3. Loudoun County Available Land Database

3.4.1 Criteria for Success

- The team creates a functioning database for NOVEC.
- The team provides NOVEC the ability to easily update the records in the database.
- The team provides NOVCE the ability to easily generate useful reports from the records in the database.
- The team creates a final report and presentation detailing the project.

3.4.2 Technical Approach

The team has divided the approach into four sections: Data Discovery, Data Retrieval, Database Design, and Report Generation.

Data Discovery
The team will search the websites provided by NOVEC to analyze what data is provided. We will communicate with NOVEC to determine what data is relevant for their future analysis. If the data is unable to be acquired from the websites, we will contact the counties directly to determine if there are alternate means to retrieve the data.

Data Retrieval
Once it is determined what data is required and what format the data is in, the team will determine the best means to retrieve the data. This could require researching a suitable scripting language or creating a manual retrieval process. We will then create the code or the method to retrieve the data.

Database Structure
Once the relevant data is determined, we can begin designing the database. We will determine the database software based on ease of use and ability to interface with NOVEC. We will then create the appropriate tables and use our data retrieval tool to populate the database. We will provide NOVEC with the ability to add comments to the records and with the ability to flag important projects.

Report Generation
Once the data is collected, we will generate reports when the data is updated. These reports will highlight new projects, prioritizing according to categories specified by NOVEC. The reports will also highlight updates to existing projects. We also give NOVEC the ability to generate reports in a format convenient to interface with their existing forecasting models.

Testing
We will be able to test the system by creating a simulated data source. We will then test whether our data retrieval tool is able to acquire the data correctly and insert the data into the database. We can then verify the data is correctly stored in the database. We can then determine whether the correct reports are generated, in terms of content and format.

3.4.3 Project Plan

3.4.3.1 Project Management
The GMU project team will engage in project management best practices in order to measure progress, ensure technical performance, and limit scope creep. Project budget is of limited concern due to the nature of the design and develop labor planned for this project. Best practices will include active schedule management, change management, and risk mitigation.

Change management will provide a method to resist scope creep. Changes will be analyzed for necessity, value, and impact. Only those changes that are necessary to meet the minimum success criteria will be approved. Members of the review board will include project team member and key stakeholders. This process is of utmost importance due to the short time period allocated for this project. Additional project management processes are described in subsequent sections.

3.4.3.2 Engagement plan

The stakeholder engagement plan is provided in Table 1. This plan will facilitate the participation of key stakeholders throughout the project development life cycle. Table 2: Stakeholder Contact Information includes the contact information for the key project team members and stakeholders.

Table 1: Stakeholder Engagement Plan

Stakeholder	Modality	Frequency	Purpose	Responsible
Project Team	E-mail Google Hangout Face to Face	Semiweekly	**Project Planning:** Communicate project schedule, progress, risks, task assignment **Technical Reviews:** Review technical design artifacts and develop solutions to design challenges.	Team Lead
Project Sponsor	Phone Email	Weekly As Needed	**Requirements Elicitation:** Generate client input and develop a better understanding of the needs of the stakeholder. **Progress Report:** Discuss project progress and challenges. Receive client feedback and relay information requests. Develop a better understanding of stakeholder needs and close existing information gaps.	Client point of contact
Faculty Advisor	Face to Face	Weekly	Discuss progress and generate feedback for improvements.	Team Lead
Faculty Review Board	Project Briefing	End of project	Present project design and development. Receive sign off for project completion.	Team Communications Officer

Table 2: Stakeholder Contact Information

Individual	Organization	Contact Information
Austin Orchard	GMU Team	Phone: 571.218.6638 Email: AJOrchard12@gmail.com
Brian Smith	GMU Team	Phone: 202.344.0721 Email: brianbsmith@gmail.com
Tygue Ferrier	GMU Team	Phone: 703.220.0311 Email: tygue.ferrier@gmail.com
Dr. Philip Barry	GMU Sponsor	Email: pbarry@gmu.edu
Bryan Barfield	NOVEC	Phone: 703.754.6723 Email: rbarfield@novec.com

Ally Shi	NOVEC	Email: XShi@novec.com
Dr. Edward Huang	Academic Advisor	Phone: 703.993.1672 Email: chuang10@gmu.edu
Dr. Abbas Zaidi	GMU Academic Advisor	Phone: 703.993.1774 Email: szaidi2@gmu.edu

Work Breakdown Schedule

1. Project Management
   - 1.1. Problem statement definition
   - 1.2. Project Proposal
   - 1.3. IPR Briefing 1
   - 1.4. IPR Briefing 2
   - 1.5. Final project briefing
   - 1.6. Risk Management plan
   - 1.7. Project Schedule
   - 1.8. Project Briefing
     - 1.8.1. Draft Project Briefing
     - 1.8.2. Final Project Briefing
   - 1.9. Project Report
     - 1.9.1. Draft Project Report
     - 1.9.2. Final Project Report

2. Systems Engineering
   - 2.1. Concept of Operations
   - 2.2. System specifications
   - 2.3. System Architecture
     - 2.3.1. Preliminary system design
     - 2.3.2. Detailed system Design
   - 2.4. Database Design
     - 2.4.1. Identify data
     - 2.4.2. Integrated data dictionary
     - 2.4.3. Develop database design
   - 2.5. User guide

3. Development
   - 3.1. Data mining application
     - 3.1.1. Research data mining
     - 3.1.2. Software design
     - 3.1.3. Code
   - 3.2. Develop Database
   - 3.3. Prototype User Interface
     - 3.3.1. Design user interface
     - 3.3.2. Develop user interface
   - 3.4. Web Site Development
     - 3.4.1. Design
     - 3.4.2. Develop

4. Integration and Testing
   - 4.1. VRTM
   - 4.2. Test procedures
   - 4.3. Qualification testing

3.4.3.3 Schedule

The major milestones planned for the NEXIS project are provided in table xx.xx. These milestones provide a framework for the deliverables and major project briefings.

Table 3: Project Milestones

Milestones	Date
Project Definition Presentation	January 29, 2015
Project Challenges Presentations	February 5, 2015
Project Proposal	February 15, 2015
In Progress Review 1	February 17, 2015
Professor Working Group Meeting	March 5, 2015
In Progress Review 2	March 19, 2015
Professor Working Group Meeting	April 2, 2015
Draft Final Report	April 16, 2015
Final Presentation Dry Run	April 23, 2015
Final Report	April 24, 2015
Professor Working Group Meeting	April 30, 2015
Final Presentation Dry Run - NOVEC	May 1, 2015
Final Presentation (5/8/15)	May 8, 2015

The NEXIS project schedule is provided in the appendix. The schedule incorporates all of the items laid out in the WBS and milestones figures. The schedule will be managed as a living document and service as a measure of project progress. The project team will be able to identify schedule risks early and develop mitigation plans by managing the schedule and tracking progress.

4.0 Concept of Operations

This section describes the proposed system in detail including background, constraints, and user classes.

4.1 Objectives

The proposed system was solicited by NOVEC to provide additional modeling capabilities. NEXIS will expand the capabilities of the current models employed by NOVEC. NOVEC has the ability to use data on ongoing development projects to refine demand forecasts. This system will enable them to easily access up to date development data through a single centrally located data store. NEXIS has four main functions; collection, processing, storage, and reporting.

4.2 Operational Policies

NEXIS is required to integrate with all existing NOVEC operational policies. See section 6.2 constraints section for more information.

4.3 Description of the Proposed System

NEXIS is an automated data collection tool that collects, stores, and reports data to the user. This system eliminates the time prohibitive task of collecting thousands of data points about ongoing construction projects by hand. NEXIS uses publicly available data from Loudoun County to populate a database and display data to users for use in predictive demand models. The system provides three high level capabilities data collection, data storage, and data reporting.

The first required capability is data collection. NEXIS is capable of parsing online sources and processing the data prior to storage. The processing will require that the data be checked for content, anomalies, and uniqueness. NEXIS first flags any missing data fields from a newly detected record and attempts to fill in missing fields with data available from previously stored records for the same project. Next the system checks for any anomalies against previous records. These anomalies are flagged for user review. Finally the system verifies that the record is unique. If a record exists that contains the identical data for the same project then the new record is not added to the database. The data processing conducted by NEXIS insures that all data stored in the system is as accurate and complete as technically feasible.

The second capability demonstrated by NEXIS is data storage. NEXIS stores all applicable data in a database for ease of access by users. The storage of collected data eliminates the need to constantly collect data from the various online sources. Additionally, storing data locally increases the speed of queries. This provides easier access to the data needed by the predictive models.

The final capability of NEXIS is the data reporting capability. The data reporting capability will provide an automated process for providing the forecasting information to the NEXIS user. The data report will be sufficiently detailed to accurately populate the forecasting model.

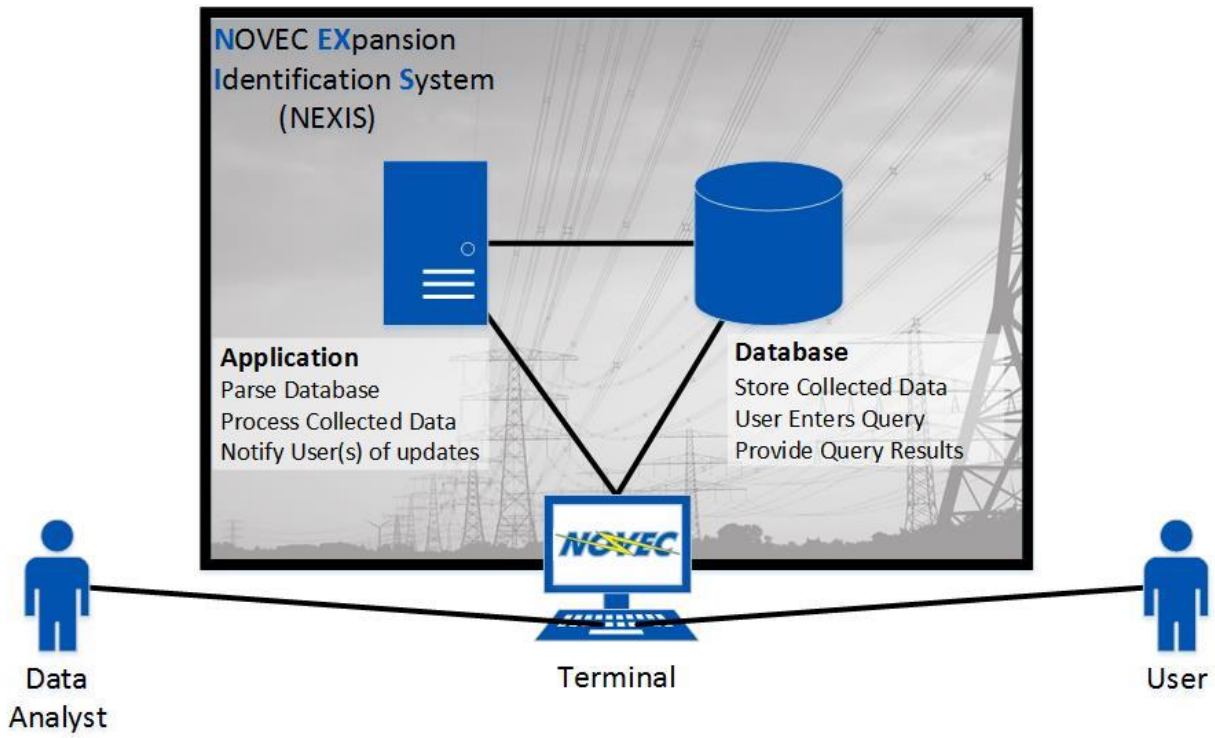


Figure 3: Notional Operational Concept

4.3.1 System Components

NEXIS is comprised of both physical components and interfaces. The physical systems and software can be divided into two main components.

Application:

Data Store:
The second component is the data store. This component provides the storage and access capabilities needed to generate reports used as model imports. The data store will consist of a database, and the hardware necessary to deploy the database. The data store hardware can be the same hardware used to deploy the web scraper.

4.4 Modes of Operation

NEXIS has three operational modes:

- Nominal
- Degraded
- Offline

Nominal describes the system condition when all functions are performed without incident. This is the normal operational mode. In the operational mode the system will collect, process, and store all identified data.

The degraded mode indicates that the collection process is currently unavailable. In this mode the system is still capable of providing access to the data and generating reports.

Offline describes the system condition when one or more components are nonfunctioning and render the system unavailable. This mode indicates that the data is inaccessible and prevents the generation of reports.

4.5 User Classes

The users for this system are considered actors who interact with the system. The user will interact with the system from the perspective of their class. The two major user classes for this system include General Users and Administrators.

General User:
This user will access the system using a username and password. They access data records and generate reports. This user does not have the ability to alter records stored in the database.

Analyst:
This user will access the system using a username and password. They access data records and generate reports. This user does have the ability to alter records stored in the database.

Administrator:
Contracted IT services data steward. Has read/write access to the database in addition to administrative functions.

4.5.1 Organizational Structure

There is no unifying organizational structure for the user base as a single person within the Project Planning organization can have one or both roles within the NEXIS process.

4.5.2 Profiles of User Classes

1. General User:
    a. Who: Non-privileged user. Typically a person who has a need to access the system without privileges.
    b. Access: Read-only access to the database
    c. Responsibility: Logging in, processing data, triggering monthly database updates, requesting updates to incorrect records
2. Analyst:
    a. Who: Privileged user. Can perform both user functions and database administrative functions. Typically a person who has extensive knowledge of the system data processing.
    b. Access: Read/Write access to the database
    c. Responsibility: Responsibility: Logging in, processing data, monthly database updates, updates to incorrect records
3. Administrator:

a. Who: Administrative user. Can perform both user functions and database administrative functions. Typically with backend knowledge that can maintain database, network and software. Third party IT support can also fill this role.
b. Access: Read/Write access to the database and administrative functions
c. Responsibility: Logging in, manually updating database, assigning privileges, system startup, system shutdown, reading log files, manually backing up database

4.6 Operational Scenarios

The NEXIS ConOps describes the capabilities desired by the NEXIS users. Subsequent sections describe the primary capabilities and success scenarios for NEXIS. This section is not intended to describe every system scenario. This section provides the main scenarios needed to reach the expectations of the main stakeholders and provide the context for developing the system design.

4.7 User Goal Use Case Diagram

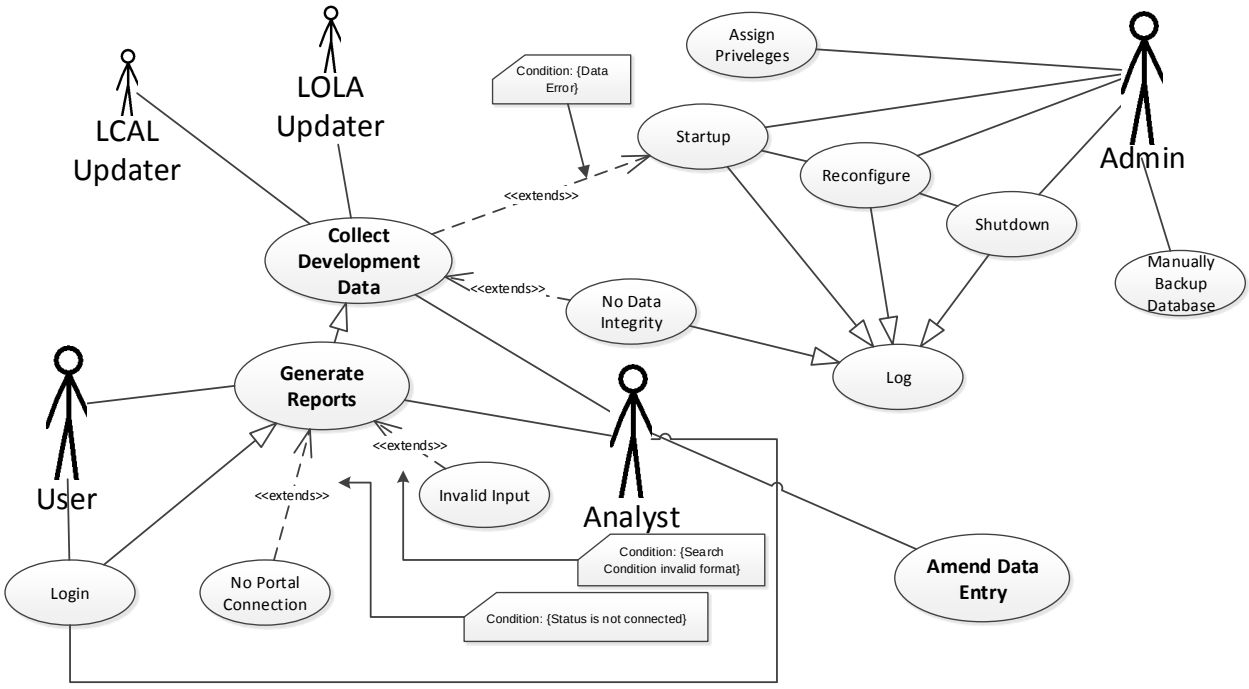


Figure 4. Use Case Diagram

The detailed descriptions of the use cases and their interaction. The end goal is to provide the user base with an automated process to update the NEXIS database, extract information from it and manually regulate the data if necessary. The nominal use cases in bold are described below. The use cases identified as in scope are “Collect Development Data” and “Generate Reports”. The use case with Analyst as the primary actor is “Amend Data Entry”.

4.8 User Goal Use Case Descriptions

4.8.1 Scenario 1: Collect Development Data

4.8.1.1 Description

This scenario describes one of the main capabilities of the proposed system, the ability to scrape data from government data stores.

This scenario is triggered by manually by the analyst user. The database updaters send the available CSV file.

The system parses the data and renders it into a standard format. This enables the data to be processed. During this step the system will compare the collected data against existing data entries. The system will flag any entries that have matching project IDs and conflicting data fields. These data fields include address, project owner, and any other field identified during the system design process. Users will be notified of any anomalous entries that have been flagged.

The system will then store all new entries in the database. The records will be formatted to meet the requirements of the database design. The system will then notify all users of any new entries that have been added to the database.

4.8.1.2 Sequence

Primary Actor(s): Analyst, LCAL Updater, LOLA Updater

Pre Condition(s):

- New database update is needed
- Analyst or Administrator performs task

Post Condition(s):

- Database are populated with all records applicable
- Report on data integrity is displayed
- Derived fields are created

Main Success Scenario:

1. Analyst sends request for database update
2. Loudoun county sends new CSV file
3. Analyst adds new CSV to repository
4. Analyst logs into system
5. Analyst enters database update command
6. System parses data
7. System creates data integrator report
8. System updates database according to data integration report
9. System processes derived fields
10. System creates “List of Records” into CSV format
11. System creates change report for database
12. System displays update report to the analyst, with the following fields
    a. Number of new records
    b. Number of new records with missing data fields
    c. Output of new records with missing data fields

d. Number of updated records
e. Number of removed records

Extensions:

1. (a) Analyst needs access
    a. Analyst logs into database
    b. System allows Analyst privileges, go to next step
1. (b) Administrator needs access
    a. Administrator logs into database
    b. System allows Administrator privileges, go to next step
2. (a) New CSV is available
    a. LOLA Updater sends CSV
    b. LCAL Updater sends CSV, go to next step
2. (b) New CSV is not available from LOLA
    a. LOLA Updater does not send new CSV, exit scenario
2. (c) New CSV is not available from LCAL
    a. LCAL Updater does not send new CSV, exit scenario
7. (a) No data integrity
    a. Application checks Project Status against older records
    b. CSV data record returns different Project Status
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (b) No data integrity
    e. Application checks Application Name against older records
    f. CSV data record returns different Application Name
    g. Application reports data as updated record to data integration report
    h. Application counts record in update report, repeat for next record
7. (c) No data integrity
    a. Application checks Application ID against older records
    b. CSV data record returns different Application ID
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (d) No data integrity
    a. Application checks Status against older records
    b. CSV data record returns different Status
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (e) No data integrity
    a. Application checks Description against older records
    b. CSV data record returns different Description
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (f) No data integrity
    a. Application checks Applicant Name 1 older records
    b. CSV data record returns different Applicant Name 1
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record

7. (g) No data integrity
    a. Application checks Applicant Name 2 against older records
    b. CSV data record returns different Applicant Name 2
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (h) No data integrity
    a. Application checks File Date against older records
    b. CSV data record returns different File Date
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (i) No data integrity
    a. Application checks Acceptance Date against older records
    b. CSV data record returns different Acceptance Date
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (k) No data integrity
    a. Application checks Completion Date against older records
    b. CSV data record returns different Completion Date
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (l) No data integrity
    a. Application checks Existing Zoning against older records
    b. CSV data record returns different Existing Zoning
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (m) No data integrity
    a. Application checks Pin Number against older records
    b. CSV data record returns different Pin Number
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (n) No data integrity
    a. Application checks Tax Map against older records
    b. CSV data record returns different Tax Map
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (o) No data integrity
    a. Application checks Latitude against older records
    b. CSV data record returns different Latitude
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (p) No data integrity
    a. Application checks Longitude against older records
    b. CSV data record returns different Longitude
    c. Application reports data as updated record to data integration report
    d. Application counts record in update report, repeat for next record
7. (q) Invalid field format

a. Application checks format of field against older records
b. CSV contains invalid format in raw database
c. Field is reported as "NULL" to data integration report
d. Error is recorded in log file
e. Application counts record in update report, repeat for next record

7. (r) Valid field format
    a. Application checks format of field against older records
    b. Field returns valid format, repeat for next record.
7. (s) No new records available to process
    a. Application finalizes integrator report
    b. Application finalizes update report, go to next step

4.8.2 Scenario 2: Generate Reports

4.8.2.1 Description

This scenario describes the capability to generate reports from the data stored in the system database. This will trigger a user query for data inside of the updated database.

This scenario is triggered by a user requesting a report from a set of predefined queries. The user first logs into the system using a valid username and password combination. Next, the user selects a report from a list of defined reports. The system then processes the request and generates a report containing the requested data. The data is formatted such that it can be ingested by legacy modeling systems.

4.8.2.2 Sequence

Primary Actor(s): Analyst
Pre Condition(s):

- User need for report
- Database update completed

Post Condition(s):

- Input parameters of made query are correct

Main Success Scenario:

1. Analyst logs in to the database
2. System allows analyst privileges
3. System returns welcome message
4. Analyst enters query in the proper format
5. System returns all applicable record values
6. Analyst stores data
7. Analyst logs out
8. Analyst processes inputs data into legacy model, exit

Extensions:

1. (a) General User logs in
    a. General user logs into database
    b. System allows General User privileges
1. (b) Administrator logs in
    a. Administrator logs into database

b. System allows Administrator privileges

2. (a) Invalid Input
    a. User enters query parameters
    b. User has entered incorrect format for query
    c. System returns “error: retry <display of correct format>” message
2. (b) Valid Input
    a. User enters query parameters
    b. System sends query to database
6. (a) Administrator logged in
    a. Administrator logs out, exit scenario
6. (a) General User logged in
    a. General User logs out, exit scenario

4.8.3 Scenario 3: Amend Data Entry

4.8.3.1 Description

This scenario describes the capability to amend erroneous data records. Analyst needs to manually update the database. This can be caused by a NULL field or other error identified by the integrity report. In addition if a user request is given to manually update a field the administrator will research and execute as appropriate.

This scenario is triggered by a user identifying an inaccurate data record. The user would request that an administrator edits the entry. The administrator would log into the system using a valid username and password combination for an account with administrative privileges. The administrator would then revise or redact the flawed data field. The system will allow the administrator to alter multiple entries during a single session.

4.8.3.2 Sequence

Primary Actor(s): Analyst, User

Pre Condition(s):

- NULL field has been raised in the data integrity check
- User has obtained proper value

Post Condition(s):

- Database record(s) are updated

Main Success Scenario:

1. User identifies inaccurate record
2. User provides corrected value to Analyst
3. Analyst access database with analyst privileges
4. Analyst issues command to update records
5. Analyst checks database record to ensure consistency
6. Analyst logs out of system, exit

Extensions:

4. (a) Analyst needs to remove record
    a. Analyst issues command to remove record, go to next step
4. (b) Analyst needs to add new record
    a. Analyst issues command to create new record

   b. Analyst issues command to fill in all applicable fields of the record, go to next step
4. (c) Analyst needs to change record
   a. Analyst issues command to change record field, go to next step

5.0 Requirements Specification

NEXIS is an automated data collection tool that collects, stores, and reports data to user. This system eliminates the time prohibitive task of collecting thousands of data points about ongoing construction projects by hand. NEXIS uses publicly available data from Loudoun County to populate a database and display data to users for use in predictive demand models.

The system provides three high level capabilities data collection, data storage, and data reporting. The requirements defined in this section define all necessary requirements for system definition from a systems design perspective. The sources used for requirements include, Stakeholder Interviews, Use Case analysis, and Concept of Operations development. The design necessary to meet the requirements derived from these capabilities is defined in section 6.0.

5.1 Functional Requirements

This section provides all of the functional requirements for the NEXIS system.

ID	Requirement Text
NXS-F-001	The system shall provide the capacity for users logging on or off.
NXS-F-002	The system shall provide no less than three types of user roles.
NXS-F-003	The system shall provide for multiple users to access each role.
NXS-F-004	The system shall provide each user with an individualized login.
NXS-F-005	The system shall provide a general user role.
NXS-F-006	The system shall provide the general user with the capability to request database updates.
NXS-F-007	The system shall provide the general user with the capability to query database.
NXS-F-008	The system shall provide the general user with the capability to generate reports.
NXS-F-009	The system shall provide an analyst role that maintains user priveleges and analyst specific priveleges.
NXS-F-010	The system shall provide the analyst role with the capability to revise records.
NXS-F-011	The system shall provide the analyst role with the capability to remove records.
NXS-F-012	The system shall provide the analyst role with the capability to add records.
NXS-F-013	The system shall provide an administrator role that maintains analyst role priveleges and administrative specific priveleges.
NXS-F-014	The system shall provide the administrator role with the capability to start the system.
NXS-F-015	The system shall provide the administrator role with the capability to stop the system.
NXS-F-016	The system shall provide the administrator role with the capability to manually backup the system.
NXS-F-017	The system shall provide the administrator role with the capability to re-deploy the software from backup.
NXS-F-018	The system shall provide the administrator role with the capability to deploy

	software patches.
NXS-F-019	The system shall provide the administrator role with the capability to read system log files.
NXS-F-020	The system shall provide the administrator role with the capability to collect system log files.
NXS-F-021	The system shall provide the administrator role with the capability to create user accounts with the following information: 1) Username 2) Password 3) Email
NXS-F-022	The system shall provide the administrator role with the capability to remove user accounts.
NXS-F-023	The system shall provide the administrator with the capability to assign priveleges to user roles.
NXS-F-024	The system shall provide the administrator the ability to assign priveleges based on capabilities needed.
NXS-F-025	The system shall provide data collection from the Loudoun County Available Land webpage.
NXS-F-026	The system shall provide data collection from the Loudoun Online Land Application webpage.
NXS-F-027	The system shall collect all data defined in the data dictionary (Table #)
NXS-F-028	Data shall be automatically collected from the specified web pages upon user initiation.
NXS-F-029	The system shall provide the collection of data at least once every 30 days.
NXS-F-030	The system shall have the capability to collect data from all websites upon user initiation.
NXS-F-031	The system shall not create new records for duplicative data.
NXS-F-032	When a collected data field is empty the system shall populate the field with available existing data from the same project.
NXS-F-033	After acquisition of the development data the system shall check for duplicative entries.
NXS-F-034	After acquisition of the development data, the system shall format the data to the data dictionary output.
NXS-F-035	After acquisition of the development data, the system shall perform a consistency check of the data against previous records from the same project, using the fields defined in the data dictionary.
NXS-F-036	The system shall display the results of user queries.
NXS-F-037	The system shall display database updates on request.
NXS-F-038	The system shall list the new data entries, after updating the database.
NXS-F-039	The system shall list all new entries with missing data fields, after updating the database.
NXS-F-040	The system shall provide generated reports in a human readable format.
NXS-F-041	The system shall generate reports containing the results of user queries.
NXS-F-042	The system shall provide a manual user backup of the system and all of its data.

NXS-F-043	The system shall provide a configurable timer of the time desired between automatic pulls.
NXS-F-044	The system shall keep a timer of time since last data pull.
NXS-F-045	The system shall automatically pull data if configured pull timer is less than time since last pull.
NXS-F-046	The system shall output changes to the database during each query with the following format: 1) Number of new records 2) Number of updated records 3) Number of removed records
NXS-F-047	The system shall send notification to the userbase with the following information in the email body: 1) Number of new records 2) Number of new records with missing data fields 3) Output of new records with missing data fields 4) Number of updated records 5) Number of removed records
NXS-F-048	The system shall allow data to be sorted by user request using the fields defined in the database dictionary.
NXS-F-049	The system shall provide saliency to reporting on projects of 50 homes or more
NXS-F-050	The system shall provide end-to-end automated data collection ability
NXS-F-051	The system shall provide scalability to the collection and storage of the predicted future service area for counties outside of Loudoun County
NXS-F-052	The system shall enforce restrictions against data altering
NXS-F-053	The system shall provide automatic audits of data updates
NXS-F-054	The system shall provide restricted user access to modify data

5.2 Architecture Requirements

The system architecture dictates that certain minimal components be implemented. The description herein is solution independent outline of the architecture.

ID	Requirement Text
NXS-T-001	The system shall operate on a personal computer with a windows operating system.
NXS-T-002	The system shall operate within a DOS-Like or Unix-Like environment.
NXS-T-003	The system shall store all collected data in a database, using either: 1) ORACLE 2) MySQL
NXS-T-004	The system shall interface with HTML webpages.
NXS-T-005	The system shall have a web based user interface.
NXS-T-006	The system shall have a custom or open source web scraper module.
NXS-T-007	The system shall have a custom consistency checking module.
NXS-T-008	The system shall be written in python programming language.

NXS-T-009	The system shall interface with NOVEC's internal network structure.
NXS-T-010	The system shall have separate interfaces to LCAL and LOLA webpages.
NXS-T-011	The system shall limit use of data to that available for public record

5.3 Operating Requirements

The system operating environment section defines the minimum software and hardware requirements for system operation.

ID	Requirement Text
NXS-O-002	The system shall require a user to provide a username and password to access the system.
NXS-O-003	The system shall lock a user account that has unsuccessfuly attempted a login more than five times.
NXS-O-004	The system shall allow query of the database.
NXS-O-005	The system shall allow storage of generated reports to the users local machine.
NXS-O-006	The system shall have the capability to populate the database with data that was previously backed up.
NXS-O-007	The system shall display login errors to the user.
NXS-O-008	The system shall report all locked usernames to the admin.
NXS-O-009	The system shall be started from a command line interface.
NXS-O-010	The system shall automatically start all processes when a start command is issued.
NXS-O-011	The system shall be shutdown from a command line interface.
NXS-O-012	The system shall automatically shutdown all processes when a shutdown command is issued.
NXS-O-013	The system shall keep a system log of the following events: 1) login attempts 2) locked accounts 3) Missed connections to externally polled websites
NXS-O-014	The system shall log all data consistency items of the following events: 1) Number of new records 2) Number of new records with missing data fields 3) Output of new records with missing data fields 4) Number of updated records 5) Number of removed records 6) Number of invalid records

NXS-O-015	The system shall log each event with the following information: a. System timestamp b. Process/Record ID c. IP, system, or record name affected d. System detailed description

5.4 Non-functional Requirements

ID	Requirement Text
NXS-N-001	The system shall have an availability of 0.999.
NXS-N-002	The system shall be capable of storing 1 Tb of development data.
NXS-N-003	The system shall use 50% or less of the total available memory.
NXS-N-004	The system shall use 50% or less of the total CPU usage.
NXS-N-005	The system shall provide the storage of no less than 10,000 individual records

5.5 Implementation Requirements

ID	Requirement Text
NXS-I-001	The system implementation shall comply with the "terms of use" of the Loudoun Online Land Application database
NXS-I-002	The system implementation shall comply with the "terms of use" of the Loudoun County Available Land database
NXS-I-003	The system implementation shall comply with the "Bylaws of the Cooperative"
NXS-I-004	The system shall comply with the fair use policies of data published at the county level
NXS-I-005	The system shall comply with NOVEC "system backup procedure"s established by NOVEC's IT support services.
NXS-I-006	The system shall comply with NOVEC "networking procedures" established by NOVEC's IT support services.
NXS-I-007	The system shall comply with NOVEC "database deployment procedures" established by NOVEC's IT support services.
NXS-I-008	The system shall be deployed with a focus on loudoun county
NXS-I-009	The system shall manage the creation of reports formatted to integrate with the legacy NIS
NXS-I-010	The system shall integrate with legacy hardware infrastructure

6.0 System Design

6.1 Constraints

The operational policies and constraints are necessary limiting factors on the systems design. Defining status of these constraints or policies is described in this section. The constraints will be

defined as within scope, by design, by implementation, or out of scope relegated to future work. This will provide a basis for the system context and system design outlined in subsequent sections.

6.1.1 NOVEC Charter

NEXIS will not violate the terms and conditions set forth in by laws of the cooperative charter. The bylaws of the cooperative charter are controlled by NOVEC and requirements were developed to reflect the policies and necessary adherence. This operational policy was managed by providing requirements to the NOVEC stakeholders that dictated management of these bylaws.

6.1.2 Service Boundary

NOVEC has a defined service area within NOVA. To provide proper forecasting, tracking new service area boundaries is a necessary input to the current models used by NOVEC Analysts. The data store used by the NOVEC analyst to define service boundaries is the shape file. This shape file is updated dynamically as service boundaries change and is internally managed by NOVEC with inputs from its customers.

The service boundary is defined as a static software component in a .shp format. The current shape file is supplied by NOVEC as a standard component to define land parcels within their service area of Loudoun County. This shape file will feed the solution domain with a criteria for determining if a land parcel identified in the LOLA database is within the service area of NOVEC. This component is a constraint on our systems design that was deemed within scope.

6.1.3 LOLA

Loudoun County Virginia maintains the LOLA data store. The NEXIS system will utilize this data. NEXIS will conform to all fair use policies of this data which is published for public record by the Loudon County Government.

The data provided by this database deals with records of Land Applications for development projects. The data is in a standard format however it is not clean. Design choices must be made that account for formatting errors. The system design will call for an agent of Loudoun County to provide a CSV including the requested fields from the LCAL database. This formatted data will be implemented in the delivered NEXIS prototype and is considered within system scope.

6.1.4 LCAL

Loudoun County Virginia maintains the LCAL data store. The NEXIS system will utilize this data. NEXIS will conform to all fair use policies of this data which is published for public record by the Loudon County Government.

The data provided by this database deals with available land and parcels that have not yet undergone applications. The data is in a standard format however it is not clean. Design choices must be made that account for formatting errors. The system design will call for an agent of

Loudoun County to provide a CSV including the requested fields from the LCAL database. However this formatted data will not be implemented in the delivered NEXIS prototype and is by design only. This data is still considered within scope of the system.

6.1.5 LCSA

The LCSA is a searchable list of data within the LOLA database. These records are updated and provide additional data fields applicable to the data store. This database was determined out of scope for the current system and will described in the future work section.

6.1.6 Current NOVEC Forecasting Models and Data Format

The format of the data was provided by the NOVEC Analyst and SME in interview format. This data format acts as a constraint on the system that will dictate design choices regarding database format so that NEXIS integrates seamlessly with the current Analyst workflow. Our system will provide input data to the NOVEC model.

A black box approach was chosen when dealing with the existing NOVEC forecasting models due to the proprietary nature of the models, see Figure 5. Blackbox Action on the NEXIS systemThe provided data format is captured in the systems design and a complete requirements specification was developed to formalize the format. This provides a fixed variable for the NEXIS database design and the transform provided by the NEXIS application need only be dictated by the format data is received from LOLA Updater and LCAL Updater.

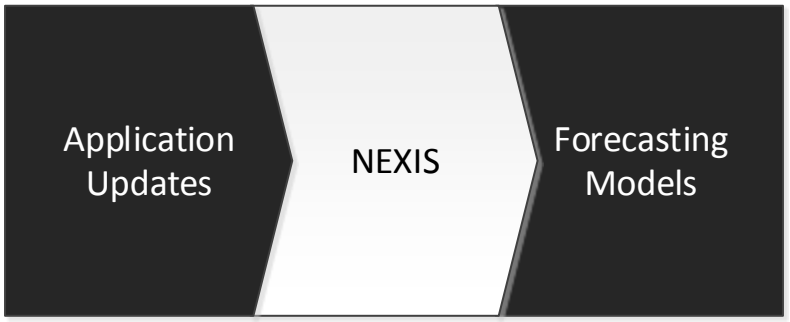


Figure 5. Blackbox Action on the NEXIS system

6.2 Capabilities Overview

6.2.1 Introduction

6.2.1.1 Architecture Overview and Summary

6.2.1.1.1 Architecture Overview

Architecture Identification

Name: SINERGY Architecture
Version: version 1.0
Completion status: Prototype
Completion date: 05/01/2015

Points of Contact

Lead Architect: Tygue Ferrier
Lead Developer: Brian Smith
Supervisor: Austin Orchard

Tools and File Formats

Tools used: Microsoft Office 2010, Microsoft Office 2013, Microsoft Visio 2010, Microsoft Visio 2013, Microsoft Visio 2013, MagicDraw v17.05, Python 2.7.9

File formats used: doc, docx, xls, xlsx, vsd, vsdx, shp, html, jpeg, pdf, py, csv

6.2.1.1.2 Architecture Models

The NEXIS architecture will be defined by 20 views namely:

Architecture Overview and Summary
Integrated Dictionary
Vision of the Architecture Capabilities
Capability Taxonomy
Capability Model Description
High Level Operational Concept Graphic
Operational Resource Flow Description
Operational Resource Flow Matrix
Operational Activity Decomposition Tree
Operational Activity Model
Operational Rules Model
System Interface Description
System Resource Flow Description
Operational Activity to Systems Function Traceability Matrix
Operational Activity to Systems Traceability Matrix
Systems Resource Flow Matrix
Services Interface Description

Service Resource Flow Description
Systems-Service Matrix
Operational Activity to Services Traceability Matrix

6.2.1.1.3 Purpose

The NEXIS architecture specification was developed to help NOVEC decision makers with decision making and allocation strategies of NOVEC resources. The architecture provides the decision maker with a solution to automate the data collection process. It will add value by providing the capability to increase situational awareness of both insipient, and in progress application status for development projects, and increase the accuracy of forecasting information by standardizing format of data and providing a data store for all operational data.

The architecture will help the development of requirements to fully implement and address the issue of data forecasting. It provides the blueprint to design, develop, and implement the system within the operating context of data acquisition

Lastly this architecture will help with the acquisition strategy to fully implement the system. The proof of concept described in this architecture will help decision makers implement a strategy of roll out to additional service areas outside of Loudoun County, up to the total service area of NOVEC.

6.2.1.1.4 Context

The history of data collection at NOVEC has been very cumbersome and inaccurate. With a black box of data the NOVEC analyst has a tough time populating the existing forecasting model using the methods outlined in their work flows. NOVEC faces the challenge of predicting energy requirements for development projects using data that is somewhat incomplete, inaccurate, and cumbersome to obtain. Existing solutions have proven ineffective of alleviating these burdens. Without fully understanding the outlets to obtain operational data directly from the County level the NOVEC forecasting models will remain incomplete.

The Environment:

The NEXIS operates with a greater operational concept with many actors operating on the system. The actors are divided into internal and external components. Figure 6 describes an overview of the data sources for the system. It will also parcel the actors on the system into groupings:

NEXIS Infrastructure – All hardware and software required to fully implement the system. It will consist of all necessary component level instantiations of the Database, and Application.

NEXIS – The NEXIS Infrastructure and user group along with the defined service boundary component, and the NOVEC leadership. Everything else within the operating context is considered an external actor and subject to the environment.

NEXIS Environment – The NEXIS and all data sources comprise the NEXIS Environment. Applicants that feed the LCAL and LOLA systems. The Land Parcel typecast are all data feeds including; Commercial Centers, Other Employment Centers, Industrial, and Residential. In

addition Loudoun County operates the LCAL and LOLA database and actors within the government structure will be the data stewards of the raw input. Loudoun County will be the sole source of operational data for the NEXIS proof of concept and prototype described in subsequent views.

External Actors – Electricity Suppliers and Northern Virginia at large comprise external actors. The electricity suppliers comprise the NOVEC customer base and are considered effected only by the forecasting models which are in turn effected by the NEXIS data format. In addition this architecture will help NOVEC determine the value of implementation across its service are in Loudoun County. The External Actors will only be affected by future works and future iterations of the system.

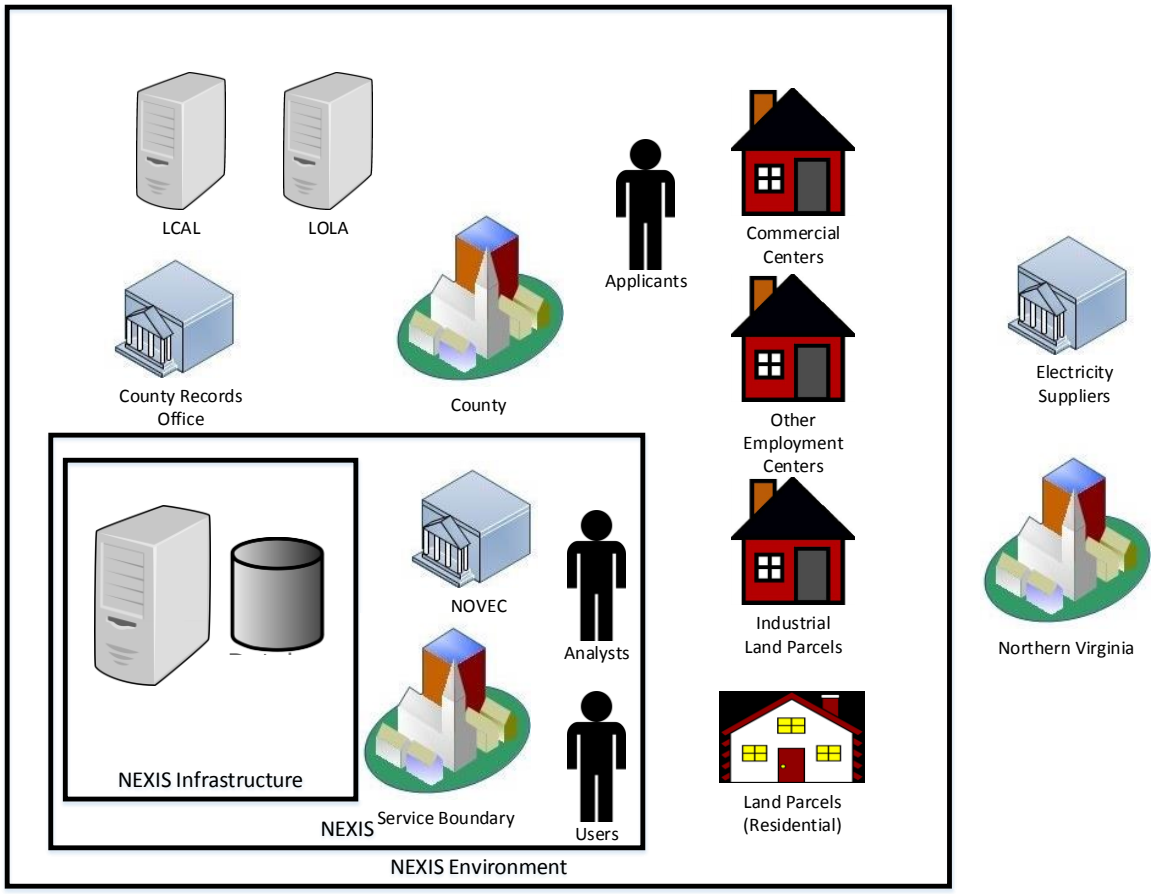


Figure 6: NEXIS Operating Context

Problem Components:
NEXIS architecture will provide a top down view that necessary decomposes the problem into two areas:

1. Situational Awareness
2. Increased Accuracy

The problem and difficulty will be realized through implementation of a system that meets these two goals. Providing the NOVEC analyst with greater situational awareness of Loudoun County and the changing status of its service areas.

Constraints:
The NEXIS architecture describes a solution that will exist within a larger operational context. As a not for profit company NOVEC is required by law to operate within the greater bylaws of the company charter and Loudoun County. The NEXIS system will adhere to all national and local data use policies. In addition all necessary interaction with Loudoun County's FOIA representation will be documented. Likewise NEXIS will operate within all fair use policies of the Loudoun County data stores.

NEXIS will take into account all operational policies and constraints that are deemed to be within scope. The operational policies and constraints include:

1. NOVEC Charter
2. NOVEC Service Boundary
3. LOLA Data
4. LCAL Data
5. NOVEC Forecasting Models
6. Data Format

6.2.1.1.5 System Release

NEXIS is developed as a proof of concept and will be presented as such. The developers of this architecture permit the reuse of data for non-commercial purposes providing full and correct attribution is provided.

NOVEC will retain control of the system and share necessary information of its operations with George Mason University and its representatives: the project team, faculty, and academic advisors. Likewise the prototype instantiation of the solution will remain proprietary to NOVEC.

6.2.2 Capabilities

Prior to description of the system, the problem statement needed to be brought into the context of our architecture and defined in a way that allowed us to define capabilities the system needed to meet. The requirements definition was used to define the capabilities section. The subsequent views describe this definition from the basis of the program vision defined in Figure 7: Vision of Architecture Capabilities. The evolution of the project vision is described by a breakout of goals and capabilities necessary to meet those goals.

These capabilities will provide the basis for development of functions and activities the system performs from the user perspective defined in the concept of operations. By mapping capabilities to functions we can see the evolution of the system design from a baseline as seen in Figure 8: Capability Taxonomy. All other views must share the goals defined in these views. Additionally NEXIS will receive greater insight into the system to help develop a formal timeline for capability deployment, as shown in Figure 9: Capability Model Description .

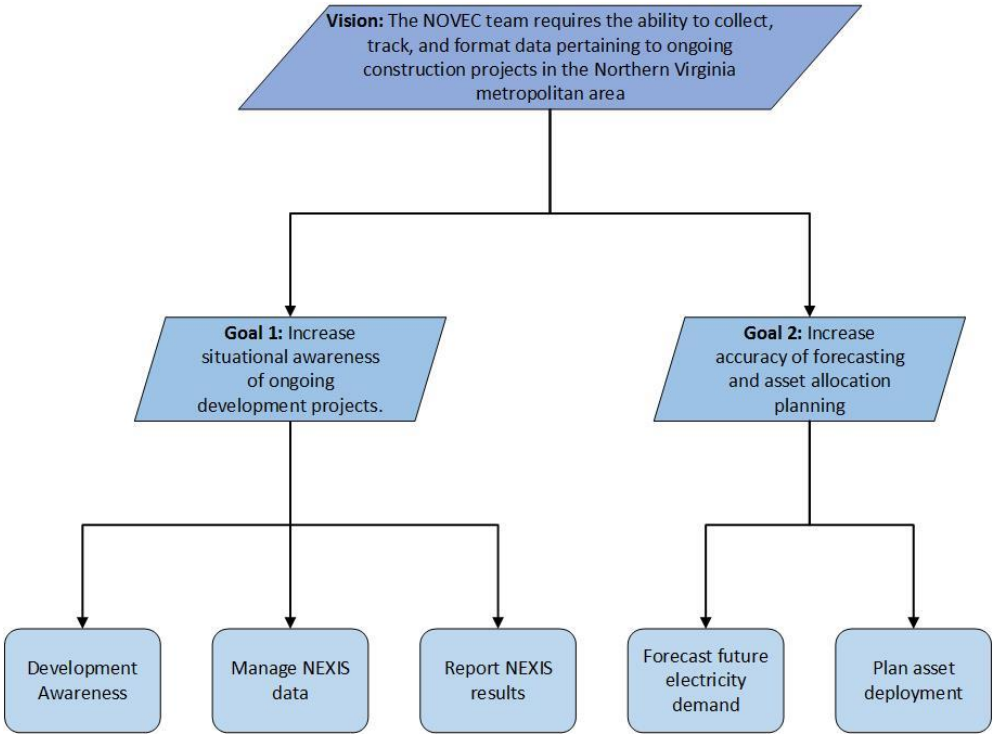


Figure 7: Vision of Architecture Capabilities

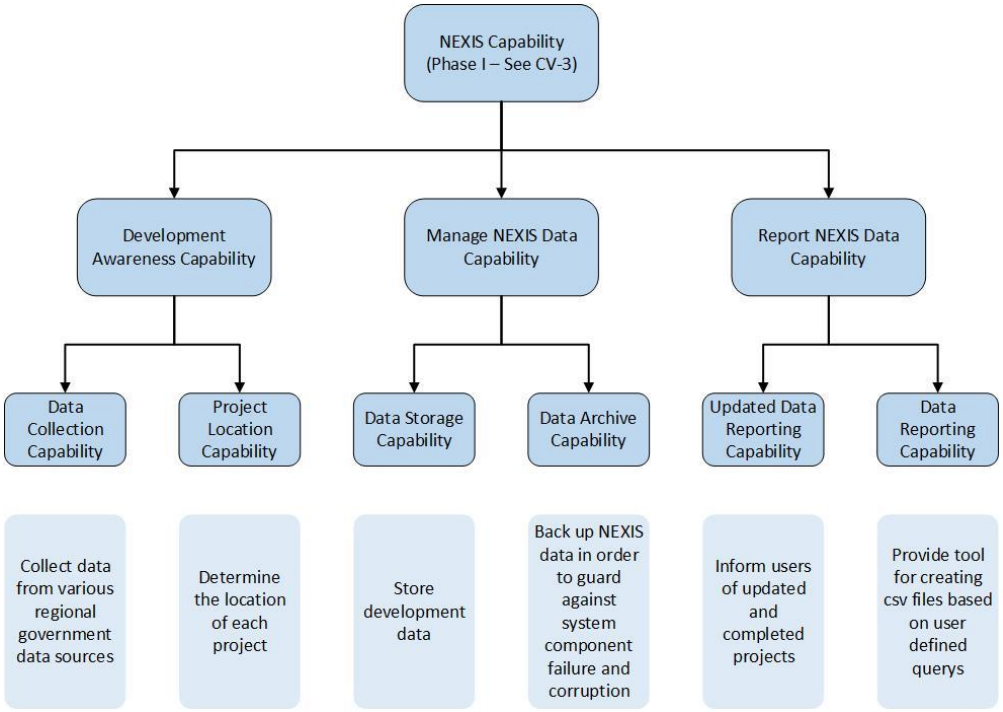


Figure 8: Capability Taxonomy

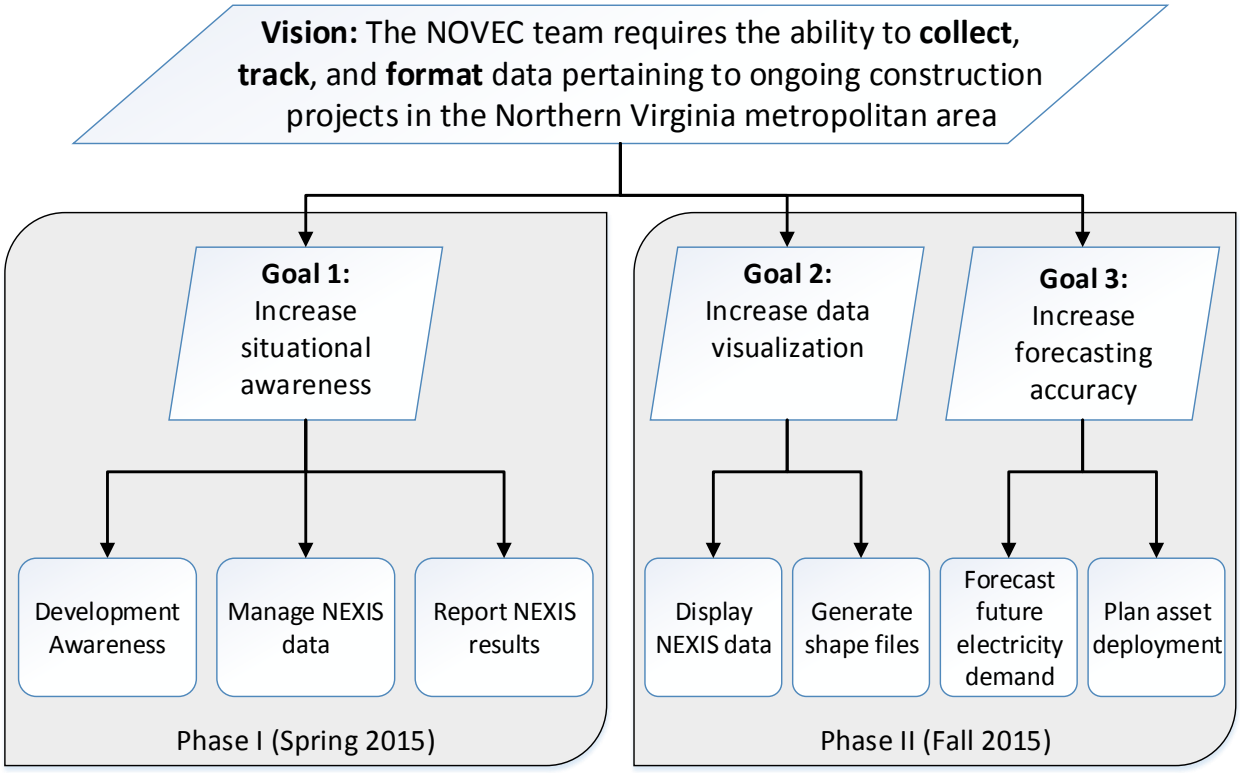


Figure 9: Capability Model Description

6.3 Operations

6.3.1 High Level Concept

Using the concept of operations and defined capabilities the high level operational concept graphic was developed, Figure 10: High Level Operational Concept Graphic. It shows the action within NEXIS on the NEXIS infrastructure from the user group and external systems within context. Subsequent definitions of the system are defined from the perspective of the NEXIS command. This diagram expands upon the concepts defined in the introduction and structures them into context.

- **External Systems:**

LCAL, LOLA, Loudoun County Application Process, Loudoun County Land

- **External Actors:**

LCAL, LOLA, Loudoun County Government, Electricity Suppliers, Applicants

- **NEXIS Command:**

Defines the boundary of the NEXIS system within the operating context. It consists of the User Group, and NEXIS infrastructure and the flow of information between the two. This action within the system is defined in subsequent descriptions.

- **NEXIS Infrastructure:**

The hardware and software solution that provide the automation capabilities defined in prior sections. This is represented as a black box and the evolution of instantiation is told in subsequent views.

- **User Group:**

Provided in context and treated as actors in subsequent views. They provide the action on the NEXIS system and will be the operational viewpoint. They will also act as operational nodes within the system context.

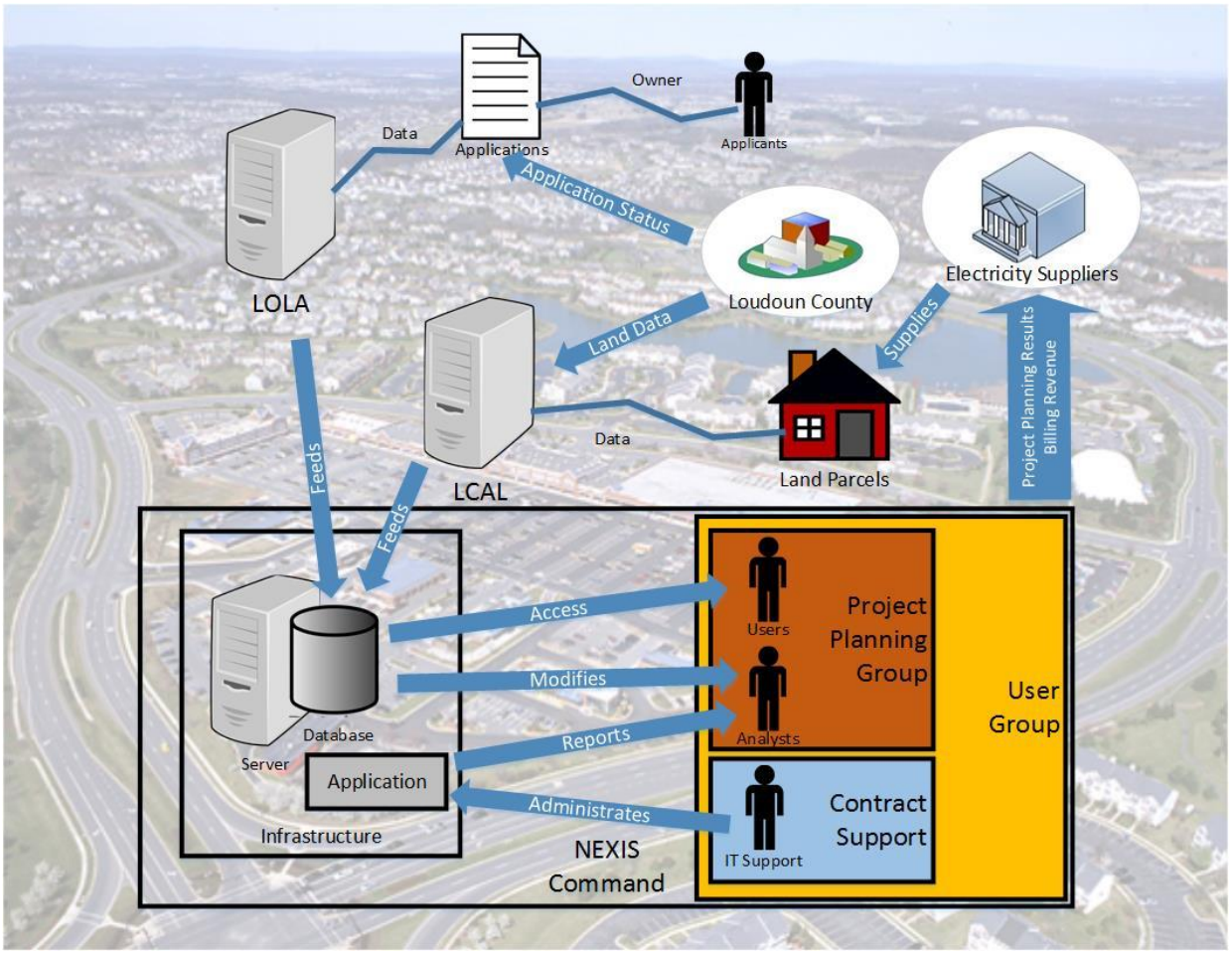


Figure 10: High Level Operational Concept Graphic

6.3.2 Resources

To start the detailed definition of the operational context, and use case diagram the resource flow between operational nodes within the system needs to be defined. Figure 11: Operational Resource Flow Description shows this action within the NEXIS system context. The resource flow makes each operational node take the form of a server and has necessary queries and

responses between nodes. The nodes within the operational resource flow will be defined as actors or components in subsequent system views.

The detailed flow and data fields within each query is defined in Table 4: Operational Resource Flow Matrix. This data format is dictated by NOVEC and was reinvestigated by the NEXIS development team after the operational view was fully defined and the system interfaces were broken out in 5.5.

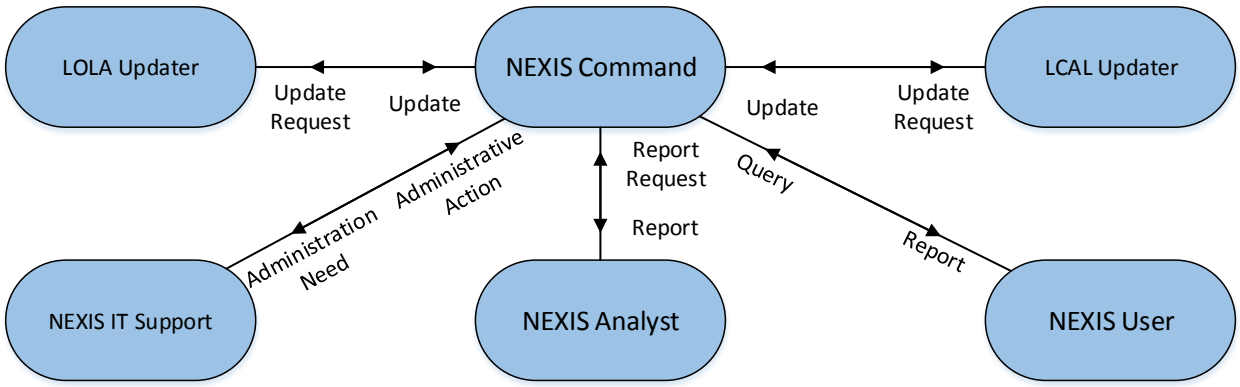


Figure 11: Operational Resource Flow Description

Table 4: Operational Resource Flow Matrix

Needline	Data	Sending Node	Receiving Node
LCAL Updater::NEXIS Command	Parcel Coordinates	LCAL Updater	NEXIS Command
LCAL Updater::NEXIS Command	Parcel Area	LCAL Updater	NEXIS Command
LCAL Updater::NEXIS Command	Parcel Type	LCAL Updater	NEXIS Command
LOLA Updater::NEXIS Command	Application ID	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Application Name	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Application Status	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Project Status	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Description	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Applicant Name 1	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Applicant Name 2	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	File Date	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Acceptance Date	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Completion Date	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Existing Zone	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Pin Number	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Tax Map	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Latitude	LOLA Updater	NEXIS Command
LOLA Updater::NEXIS Command	Longitude	LOLA Updater	NEXIS Command
NEXIS Command::LCAL Updater	Update Request	NEXIS Command	LCAL Updater
NEXIS Command::LOLA Updater	Update Request	NEXIS Command	LOLA Updater
NEXIS Analyst::NEXIS Command	Report Request	NEXIS Analyst	NEXIS Command
NEXIS Analyst::NEXIS Command	Record Update	NEXIS Analyst	NEXIS Command
NEXIS Command::NEXIS Analyst	Parcel Coordinates	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Parcel Area	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Parcel Type	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Application ID	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Application Name	NEXIS Command	NEXIS Analyst

NEXIS Command::NEXIS Analyst	Application Status	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Project Status	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Description	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Applicant Name 1	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Applicant Name 2	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	File Date	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Acceptance Date	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Completion Date	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Existing Zone	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Pin Number	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Tax Map	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Latitude	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Longitude	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Update Request	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Update Request	NEXIS Command	NEXIS Analyst
NEXIS Analyst::NEXIS Command	Request	NEXIS Analyst	NEXIS Command
NEXIS Analyst::NEXIS Command	Record Update	NEXIS Analyst	NEXIS Command
NEXIS Command::NEXIS Analyst	Report	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Project Size	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Project Demand	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Null Records	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Change in Application Status Count	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS Analyst	Change in Pre-Application Land Parcels	NEXIS Command	NEXIS Analyst
NEXIS Command::NEXIS IT Support	Administrative Request	NEXIS Command	NEXIS IT Support
NEXIS IT Support::NEXIS Command	User Update	NEXIS IT Support	NEXIS Command
NEXIS IT Support::NEXIS Command	Record Update	NEXIS IT Support	NEXIS Command
NEXIS IT Support::NEXIS Command	Start Command	NEXIS IT Support	NEXIS Command
NEXIS IT Support::NEXIS Command	Stop Command	NEXIS IT Support	NEXIS Command
NEXIS IT Support::NEXIS Command	Configuration	NEXIS IT Support	NEXIS Command

NEXIS User::NEXIS Command	Query	NEXIS User	NEXIS Command
NEXIS Command::NEXIS User	Database Dump	NEXIS Command	NEXIS User
NEXIS Command::NEXIS User	Query Report	NEXIS Command	NEXIS User

6.3.3 Operational Activities

Using our use cases, capabilities, and operational concept several operational activities were defined for NEXIS. The operational activities are defined below in four bins: Operate NEXIS, which comprises all action necessary to initiat reporting from the interface; Manage NEXIS, which comprises all activities to manage the datastore; Process Data, which includes all automation tasks provided by the NEXIS software; and Store data is inherent to the database system and its capabilities.

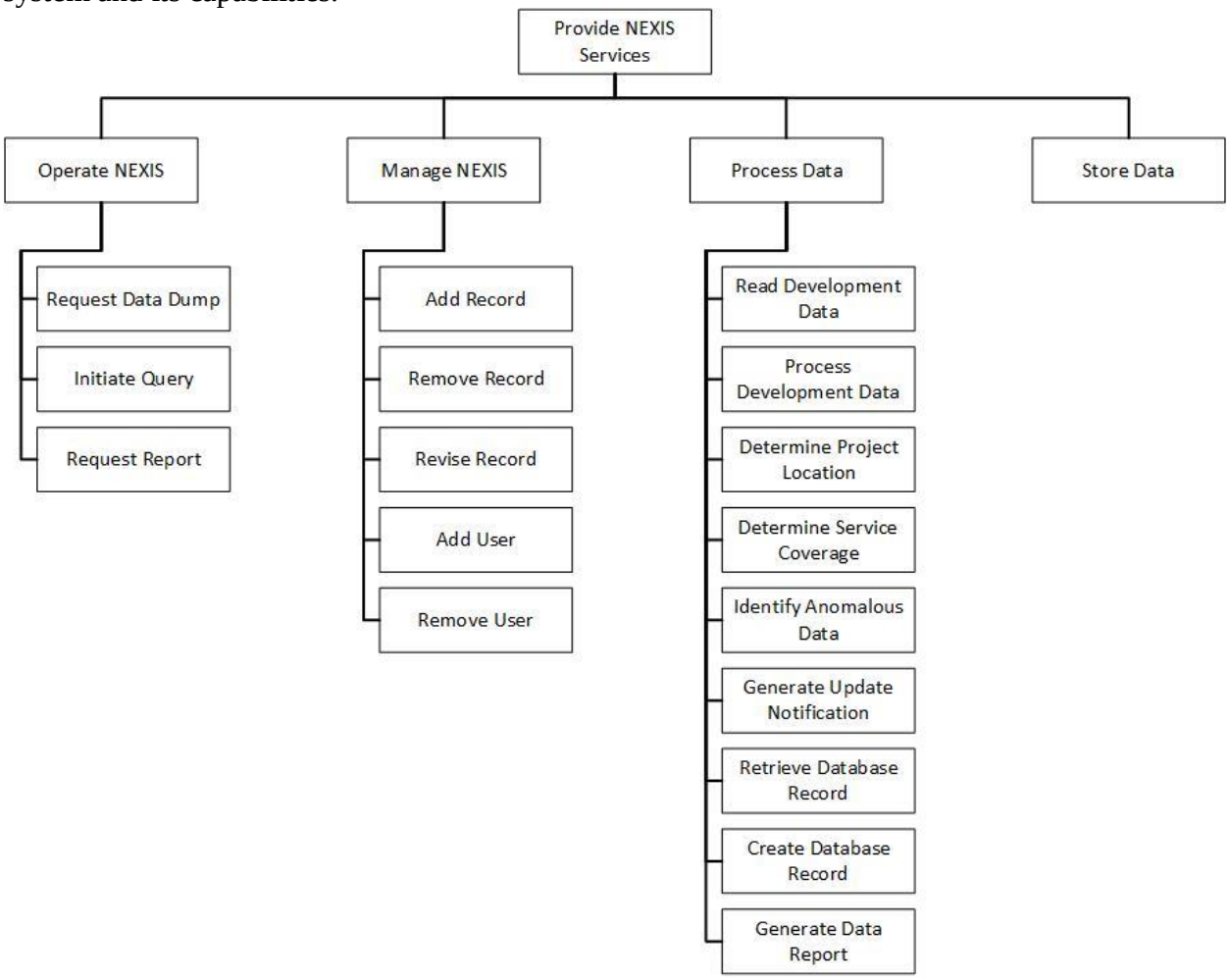


Figure 12: Operational Activity Decomposition Tree

Figure 13 and Figure 14 depict the actives mapped to systems in order of occurrence. This depiction provides the process flow from event trigger to final activity. Figure 13 depicts the update database process. The process begins with the user requesting a file containing the data in the land development database. Subsequently the user will initiate the update using the new file. The data processing application then reads and process the data. During this process the application generates the database update notification for users. This update includes both new and modified records.

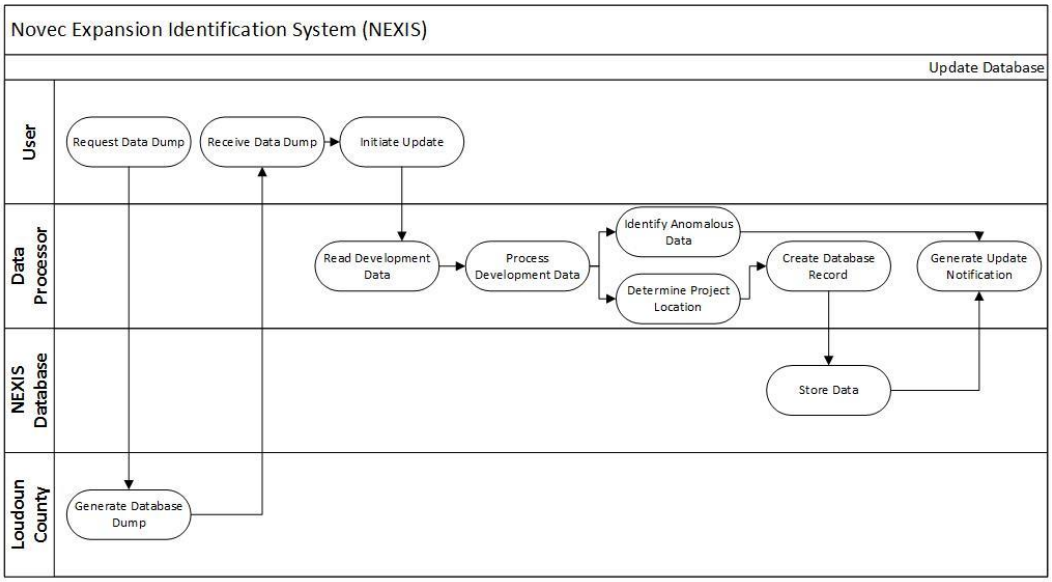


Figure 13: Update Database Activity Diagram

Figure 14 depicts the generate report process. This flow diagram represents the activities needed for the system to generate a report containing the requested development data. These reports are used as input into forecasting models. One important activity in this process is the "Determine Service Coverage" activity. This activity determines whether a project is within the NOVEC service area. Projects outside of NOVECs service area do not directly impact the modeling parameters.

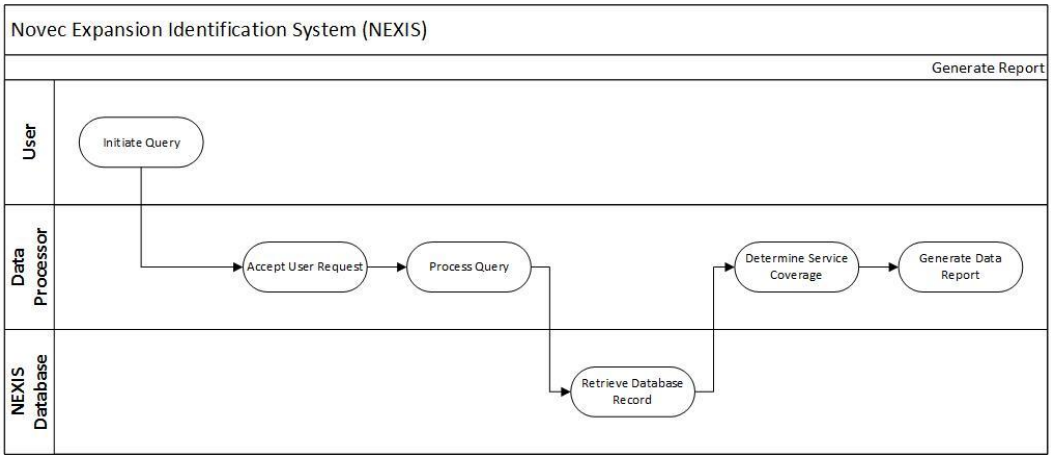


Figure 14: Generate Report Activity Diagram

6.4 Systems and Services

6.4.1 System Interfaces

The Interface description is broken up into two separate views. Figure 15: System Interface Description shows the breakout of system level interaction within the NEXIS enterprise. Connected by internet protocol the Data processing system and Database System are connected to the reporting system. This diagram describes the white box/black box action on the system in

further detail and starts the solution instantiation effort. For further information and breakout into the solution domain the necessary interfaces for these systems to interact need to be described.

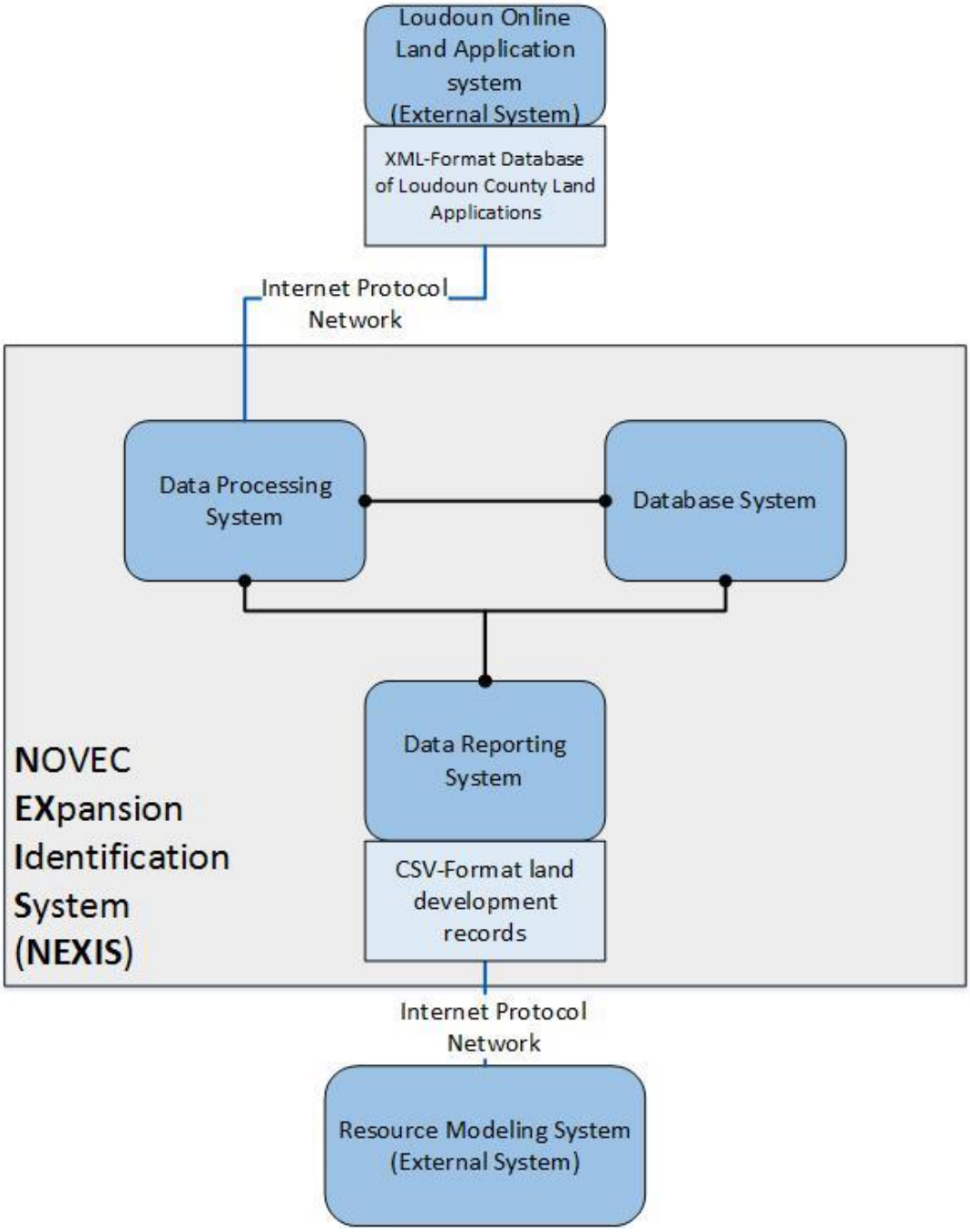


Figure 15: System Interface Description

The system interfaces can be described in a protocol stack model. Figure 16: Services Interface Description shows this model. Layers in the model are described in increasing level of user interaction from a bottom up format. The layer description are as follows:

- Layer 0 – Raw data updates
- Layer 1 – Compiled data
- Layer 2 – Internet protocols. These serve the function of abstracting the access between systems. It presents the possibility that the database system and reporting system need not be accessed from the same locally run machine.
- Layer 3 – The data reporting system will be run from a web based technology instantiated in the solution domain. The database system and technology will be described in this layer.
- Layer 4 – The application layer starts on layer 4 and consists of all data processing accessed from the previous layers. It will also house the interfaces for user interaction.
- Layer 5 – The NEXIS will provide a reporting capability. This capability will be provided locally once the database has been parsed and updated. The described interfaces are all visible to the user.
- Layer 6 –All NEXIS administrative actions will be locally run. All users regardless of privilege will need to access this layer before performing any other action on the system.

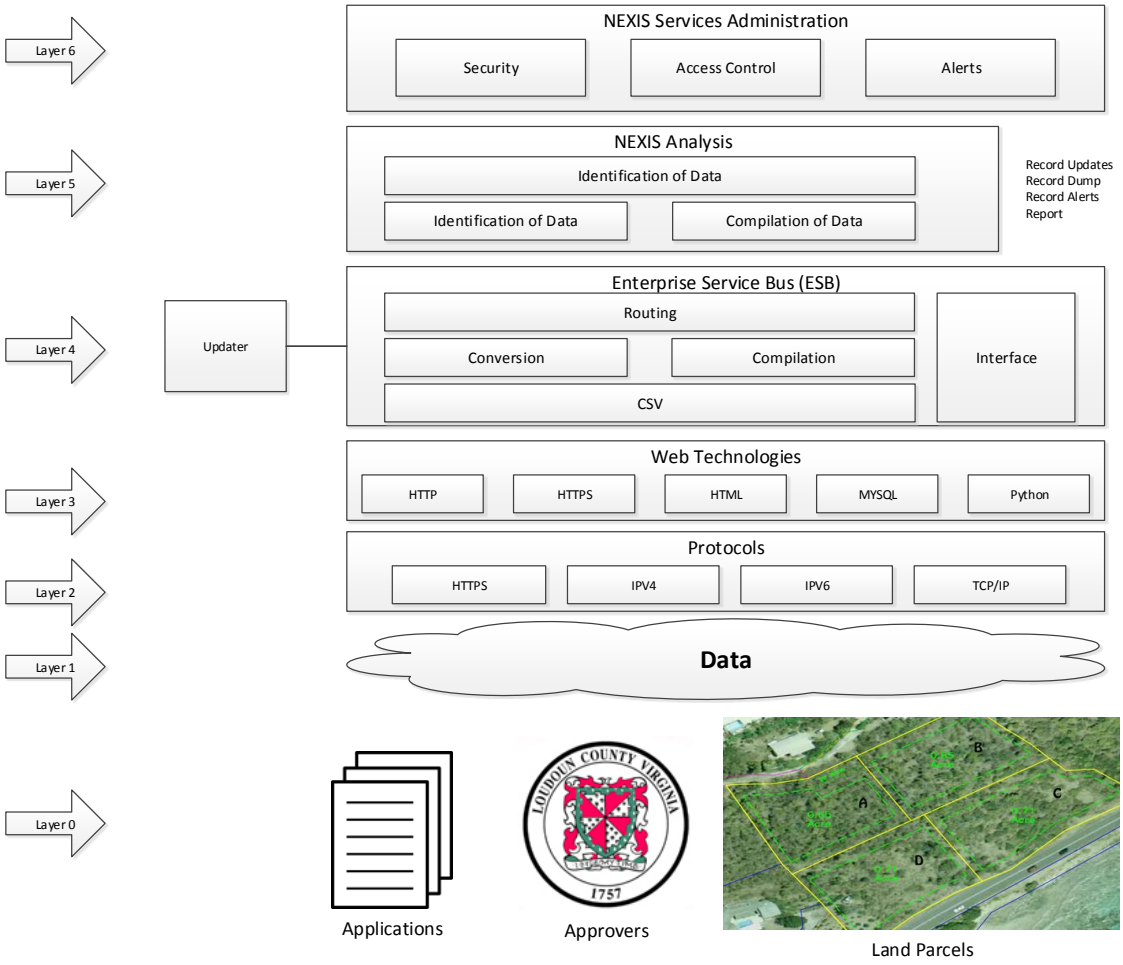


Figure 16: Services Interface Description

6.4.2 System Services

Building upon previous functional and system interface description, the component level interaction with the user must be described. Figure 17: Service Resource Flow Description shows a resource flow between the operational nodes of the system. The nodes are described in two formats: actors and components. The command center, Analyst, General User, and Administrator all comprise actors. To find the operation description of the actors on the system see section 4.8.

The components are the itemized components necessary to build the system. The solution domain will be instantiated within the parameters of this description. All components will provide the necessary services to provide the intended capabilities of the system. The components are Interface Component, Data Integration Component, Operation Data Store Component, Security Component, and Notification Component.

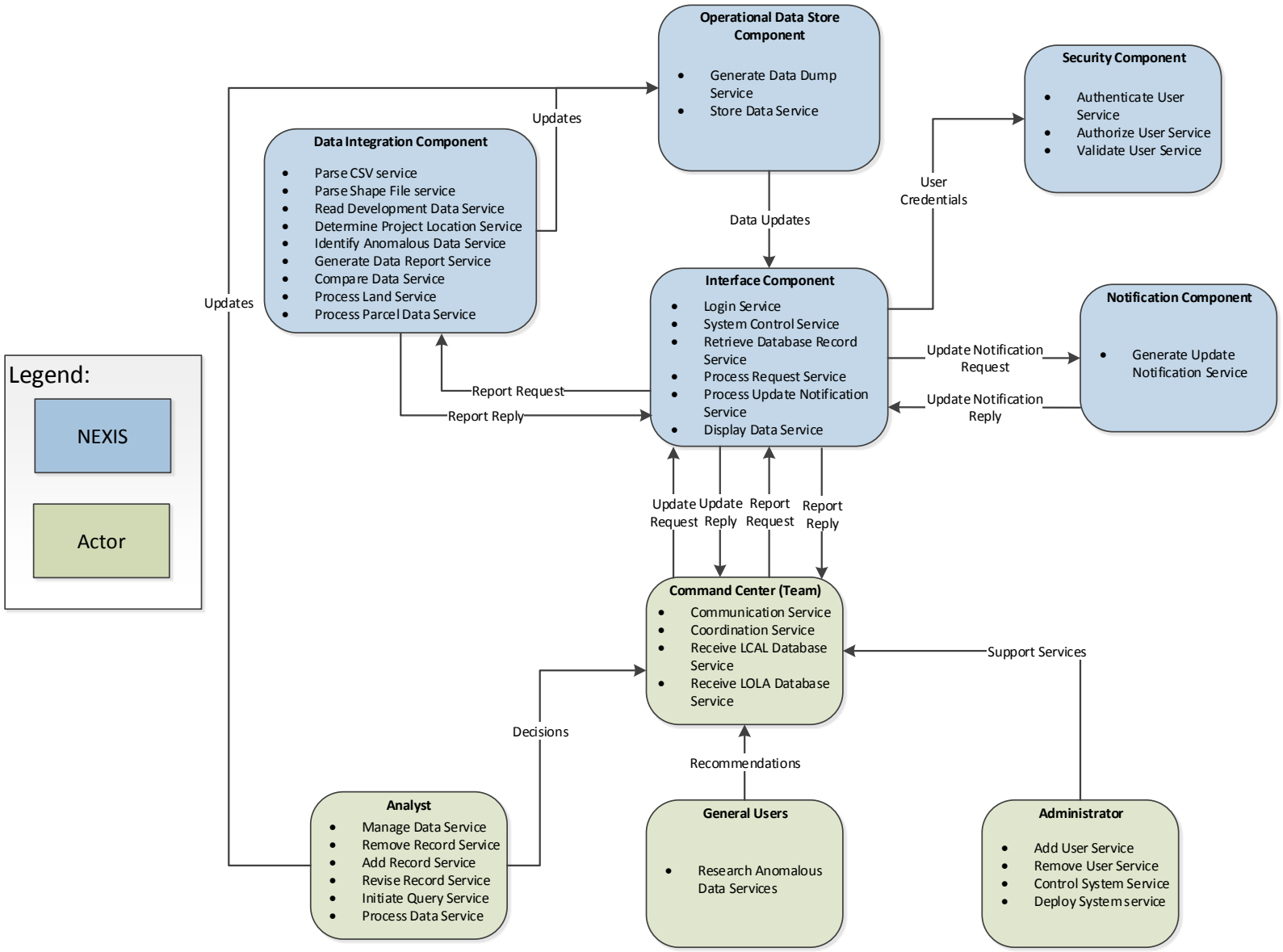


Figure 17: Service Resource Flow Description

6.4.3 Solution Instantiation

The trade spaces below reflect the analysis conducted to determine the component level instantiation of the system.

User Interface Component
The User Interface component will provide the services necessary to provide the user with functional access to the system's other components and services.

Technology Choices:

	Intended Benefits	Risks
GUI	- Greater usability - More powerful representation - Greater customization	- More development time needed - Modules can get very large - Hindered processing time
Command Line	**- Easy to implement** **- Fully functional and integrated into windows already**	**- More difficult to expand**

Data Integration Component
The data integration component will parse the data gained from the LOLA and LCAL databases, prior to storage. Its actions are not directly visible to the user but are logged with the logging capability. It will provide the operational data store with pre-processed data and the notification component with the update counts.

Technology Choices:

	Intended Benefits	Risks
Python (v2.7)	**- Self-Documenting** **- Easy to read and implement** **- Expandable** **- Easily deployed modules for reading shape files**	**- Not as powerful as other choices** **- Modules can get very large**
Java (v7.1)	- More powerful - Most number of modules - More robust support - Industry Standard - Better data visualization	- Difficult to implement within cycle - More difficult to expand

Operational Data Store Component
The operational data store component will provide the services necessary to house the database and manage the data. It will interact with the interface component and provide the notification component with raw data for reporting services.

	Intended Benefits	Risks
MySQL	**- NOVEC currently uses** **- Free** **- Easier to learn**	**- Not as powerful as other choices**

Oracle	- More powerful - More flexibility - Greater expandability - Industry Standard - Better visualization	- Difficult to implement within cycle - Greater cost - More difficult to implement

Security Component
The security component will provide the services necessary to conduct security capability such as the login functionality. It will provide

	Intended Benefits	Risks
Linux (CENTOS 6.1)	- Easy to implement	- Not as intuitive - Unnecessary to implement
Windows (Windows 7)	**- NOVEC currently uses - Interacts with NOVEC infrastructure**	**- Difficult to implement within application**

Notification Component
The Notification Component will provide the services necessary to process the counts provided by the data integration component. It will then provide the report on database changes to the user piped through the interface.

Technology Choices:

	Intended Benefits	Risks
Email	- Access is easy - Better usability	- Difficult to develop and configure within project cycle
Direct Reporting	**- Easy to Implement - Little development required**	**- Not as easy to read (can be misinterpreted)**

6.4.4 System Traceability

After system architecture and solution instantiation it is important to prove traceability to the design. To prove traceability within the system definition the services were used as the hub for verification of the systems design. We mapped the services to both the systems that provide them Table 5: Systems-Services Traceability Matrix, and the operational activities that perform them Table 6: Operational Activity to Services Traceability Matrix.

Table 5: Systems-Services Traceability Matrix

	Loudoun Online Land Application System	Loudoun County Available Land System	Data Processing System	Database System	Data Reporting System	Security System	Resource Modeling System	Command System
Process Land Service			x					
Parse Shape File Service			x					
Read Development Data Service			x					
Determine Project Location Service			x					
Identify Anomalous Data Service					x		x	
Generate Data Report Service					x			
Process Land Data Service			x					
Process Parcel Data Service			x					
Generate Data Dump Service					x			
Compare Data Service			x					
Store Data Service				x				
Login Service						x		
System Control Service			x					
Retrieve Database Record Service				x				
Process Request Service			x					
Process Update Notification Service					x			
Authenticate User Service						x		
Authorize User Service						x		
Validate User Service						x		
Generate Update Notification Service					x			
Communication Service								
Coordination Service			x					x
Add User Service								x
Remove User Service								x
Control System Service			x		x			x
Deploy System Service								x

Research Anomalous Data Services								x
Manage Data Service								x
Remove Record Service				x				x
Add Record Service				x				x
Revise Record Service				x				x
Initiate Query Service								x
Process Data Service			x				x	
Receive LCAL Database Update Service		x	x					
Receive LOLA Database Update Service	x		x					

Table 6: Operational Activity to Services Traceability Matrix

	Request Data Dump	Initiate Query	Request Report	Add Record	Remove Record	Revise Record	Add User	Remove User	Read Development Data	Process Development Data	Determine Project Location	Determine Service Coverage	Identify Anomalous Data	Generate Update Notification	Retrieve Database Record	Create Database Record	Generate Data Report	Store Data
Process Land Service										x	x						x	
Parse Shape File Service									x			x				x		x
Read Development Data Service									x	x							x	
Determine Project Location Service											x							
Identify Anomalous Data Service													x					
Generate Data Report Service														x				
Process Land Data Service										x								
Process Parcel Data Service										x								
Generate Data Dump Service																	x	
Compare Data Service													x	x			x	
Store Data Service																		x
Login Service	x	x	x	x	x	x	x	x										
System Control Service	x	x	x	x	x	x												
Retrieve Database Record Service									x						x			
Process Request Service			x						x									
Process Update Notification Service									x					x				
Authenticate User Service	x	x	x					x										

Authorize User Service	x	x	x					x										
Validate User Service	x	x	x					x										
Generate Update Notification Service														x				
Communication Service																		
Coordination Service									x									
Add User Service							x											
Remove User Service								x										
Control System Service									x							x	x	
Deploy System Service							x											x
Research Anomalous Data Services													x					
Manage Data Service				x	x	x			x							x		x
Remove Record Service					x											x		x
Add Record Service				x												x		x
Revise Record Service						x												x
Initiate Query Service		x																
Process Data Service										x	x			x	x			
Receive LCAL Database Update Service				x					x		x	x	x	x	x	x	x	x
Receive LOLA Database Update Service				x					x	x			x	x	x	x	x	x

6.5 Logical Data Design

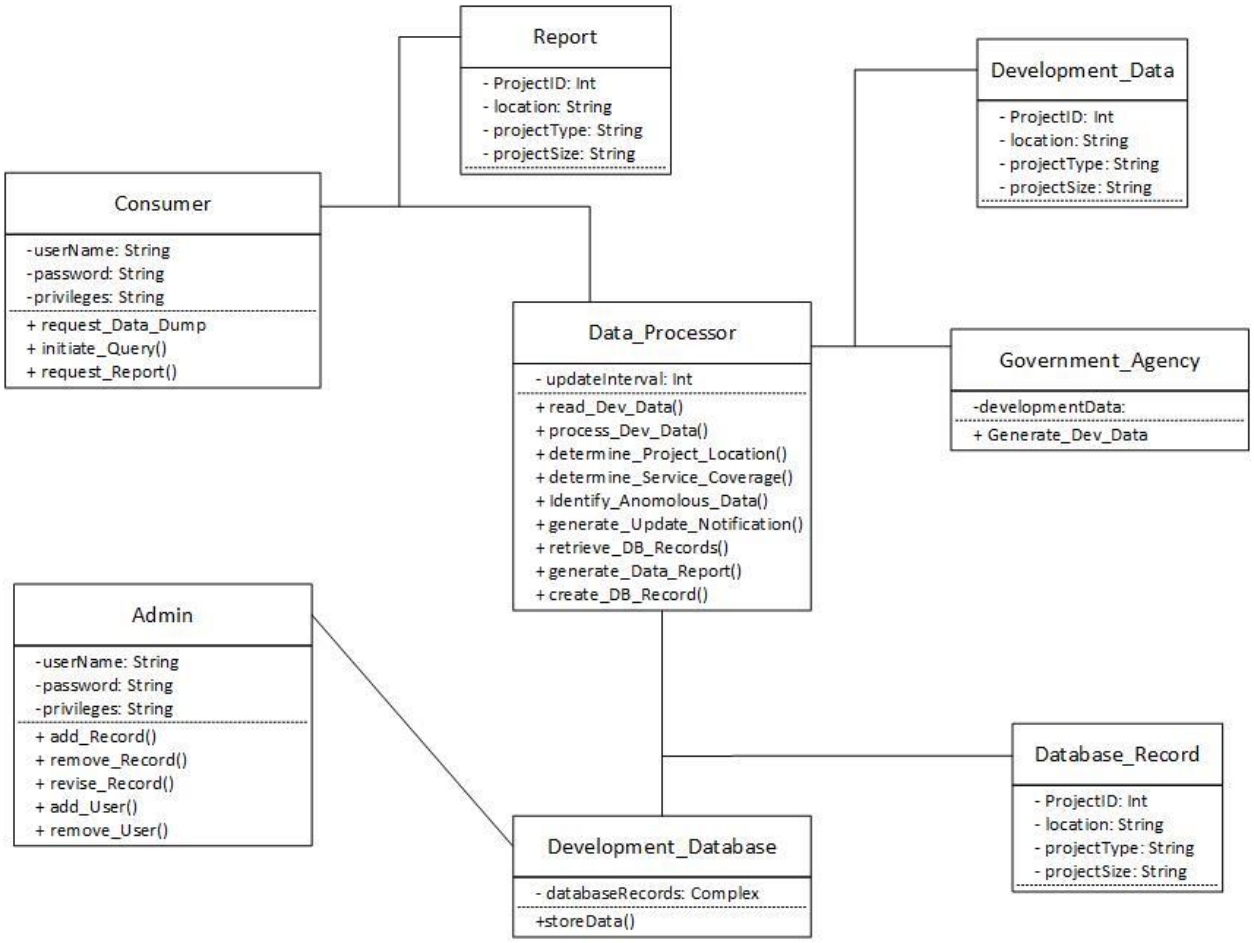


Figure 18: Class Diagram

6.6 Software Development

6.6.1 Database Design

The database was designed using MySQL. The goal in the database design was to create a structure that efficiently stores the data from the different Loudoun County Databases while also being easily expandable to other Counties. After reviewing the data available on each of the websites, we determined with NOVEC which information was relevant and required for their use. We then attempted to identify overlapping data structures that could share a common table. For example, in Loudoun's records, all three types of data had an applicant name but only the Available Land Database records had an associated phone number. For data verification, we created static data tables. Once the database is populated, those static tables can be used to check whether there are errors in the records provided during updates.

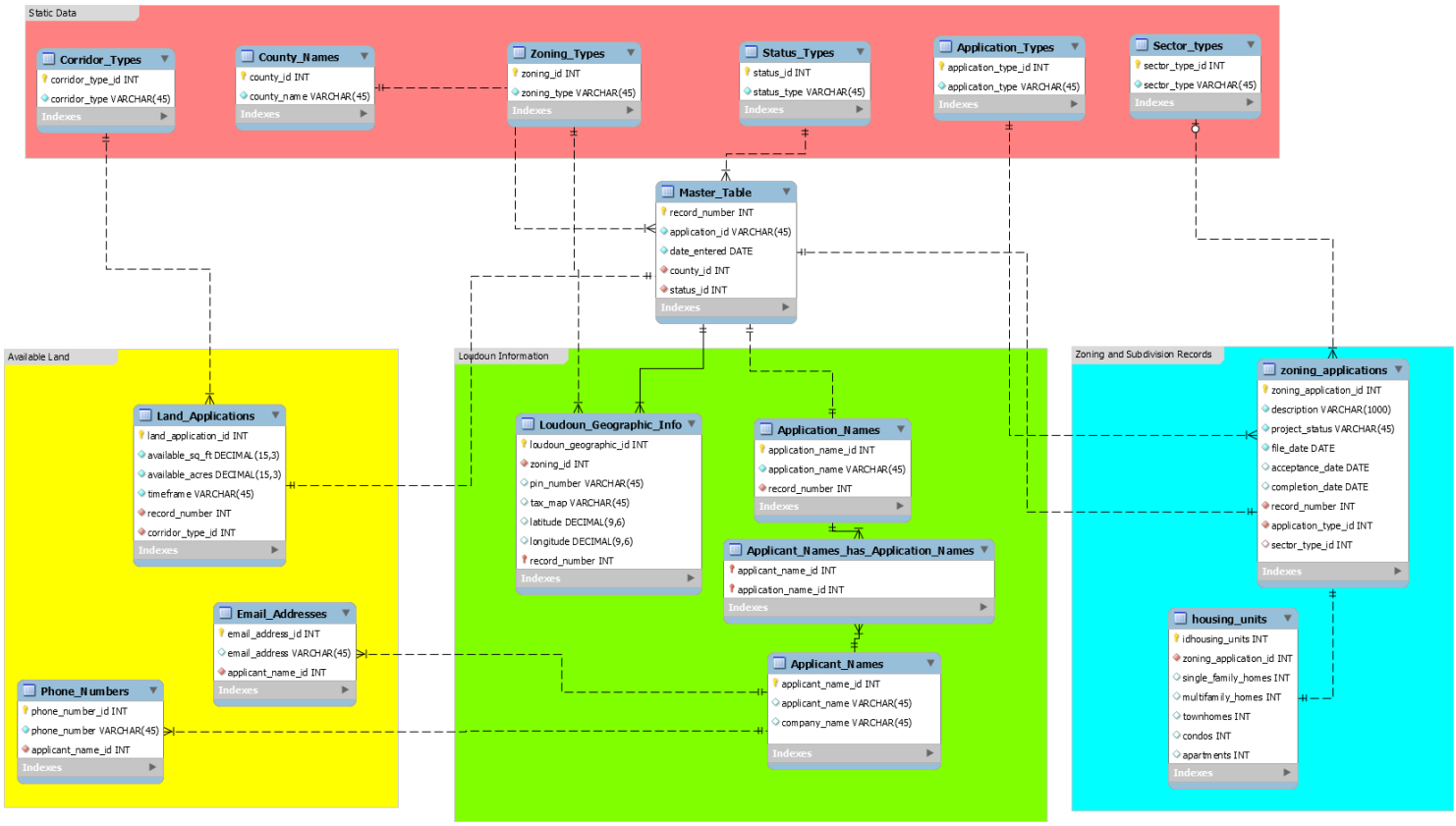


Figure 19: Database Design

6.6.2 NEXIS Development

NEXIS was designed using Python 2.7. Python was chosen for its readability, ease of use, portability, and well documented and diverse libraries. We designed a command line interface because priority was placed on function rather than aesthetics. The goal in designing the code was to create a clear structure that can be copied and modified as other counties are added. To that end, we created a LOLA Record Class to store the information, a LOLAupdater module to process updates from a csv file and a LoudounReports module to generate the different reports

requested. For more information on how to use NEXIS, refer to the NEXIS user's manual included in Appendix 8.3. For more information on the source code for NEXIS, please refer to the project webpage at http://seor.gmu.edu/projects/project_S15.html

7.0 Conclusion

7.1 Impact

The Impact of system capability implementation can be described by two measures, time savings, and increased data accuracy in comparison to the status quo. To illustrate this point an experiment was conducted. Data was scraped manually from the LOLA database in a similar manner to the NOVEC analyst. Then this data was manually analyzed to determine whether the parcel from the record was within service area.

Once data was corrected, on average it was found that errors were made in three out of the five attempts for determining service area. This reflects data that would have to be reprocessed. The total time of this process was compiled and it was found that impact on savings can be expounded per record as shown in Table 7: Experiment Data. A nominal time saving simulation can be conducted to determine how much time savings can be had with more records resulting in more time savings, Table 8: Time Savings.

Simulation Assumptions:

- Processing time for system to process up to 100 records is 1 minute
- System is 99% accurate
- There is no learning curve for compiling data or making less errors
- No other data errors exist other than service area

Table 7: Experiment Data

Number of Records	Errors	Errors per Record	Processing Time	Processing Time Per Record
5	3	0.6	28 minutes	5.6 minutes
10	5	0.5	45 minutes	4.5 minutes
Average		**0.55**		**5.1 minutes**

Table 8: Time Savings

Number of Records	Nominal Errors	System Errors	**Error Reduction**	Nominal Processing Time (min)	System Processing Time (min)	**Time Savings (min)**
5	2.8	0.0	**2.8**	25.5	1	**24.5**
50	27.5	0.5	**27.0**	255.0	1	**254.0**
100	55	1.0	**54.0**	510.0	1	**509.0**

7.2 Future Work

Increase Data Visualization:
Introduction of the capability to display NEXIS data and generate shapefiles of projects being undergone would be of value to the analyst. This would improve form of the data and would incorporate heuristics and human factors for data presentation. Stakeholder engagement and feedback would be needed to fully instantiate a solution but a possible example is expressed in Figure 20: Data Visualization Prototype.

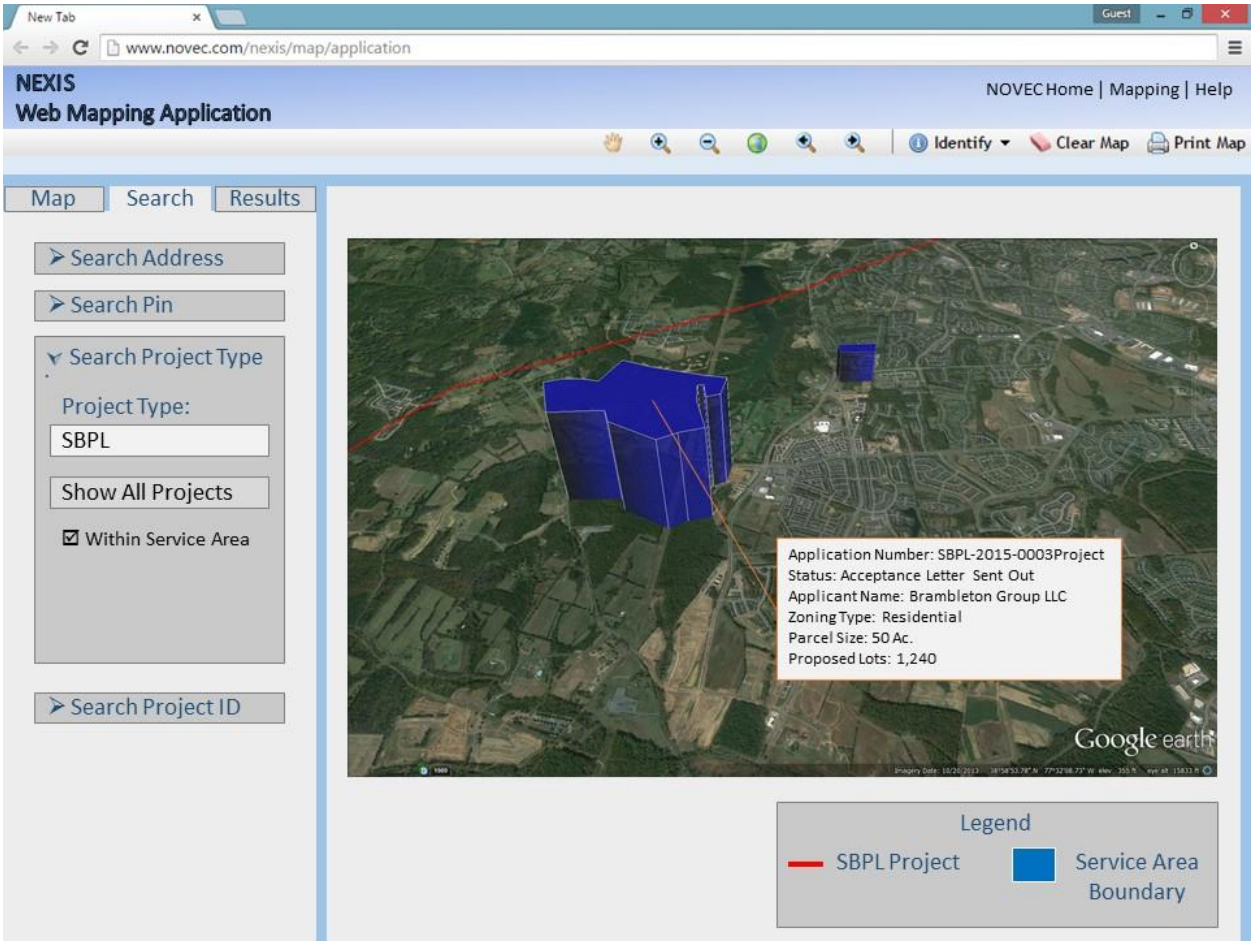


Figure 20: Data Visualization Prototype

Increase Forecasting Accuracy:
The capability to Forecast Electricity Demand and Plan Asset Deployment would add value to the effort already conducted by allowing the user to more accurately populate their models with post processed data. The data provided by the prototype system would be a direct input to this added capability.

Expand Proof of Concept:
Expansion to the 8 other districts NOVEC serves would add value by compounding the impact of the NEXIS system as implemented to the 8 other districts served by NOVEC. Addition of the capabilities described prior may increase the value of system expansion.

7.3 Concluding Remarks

This report presents the culmination of a semester long effort to generate two work products for two very different stakeholders. A final report and the NEXIS system. A closing analysis on the effort is included below.

7.3.1 Summary

This report fully documents the entire effort from a systems engineering perspective. A proposal and subsequent concept of operations was created. Stakeholder buy-in of these work products, allowed requirements to be developed and a working systems design to be instantiated. The systems design documentation was a direct feed to the delivered system and fully documents the life of this effort.

Using the design documentation the system was also built and delivered to the customer. The system was built to introduce all necessary capabilities and relegated additional functionality to future work. Additionally, an analysis on the impact of implementation was generated to illustrate the effect of system implementation over the status quo. The savings in the form of time and accuracy of data mining are explicit.

This study was generated to meet four measures of success. After determining the impact as explicit all of the measures of success for the effort were deemed as met:

- The team creates a functioning database for NOVEC.
- The team provides NOVEC the ability to easily update the records in the database.
- The team provides NOVEC the ability to easily generate useful reports from the records in the database.
- The team creates a final report and presentation detailing the project:

7.3.2 Lessons Learned

The first learnt by the NOVEC team was the critical role played by stakeholder engagement. The NOVEC sponsors were consulted throughout the life cycle of the project. This provided a collaborative forum for the exchange of ideas. The collaboration between stakeholders and the design team elicited additional client needs, providing a holistic view of the desired system. Without this dialogue the full scope of the desired system would not have been ascertained.

The design team received client generated comments through the process of formally supplying deliverables. These comments lead to further improvements on the system design. Additionally, this process ensured stakeholder buy in on each artifact. There is no substitute for consistent stakeholder engagement. This experience provided the project team with invaluable experience that will be applied to all future endeavors.

The skills to overcome the challenges of a regionally diverse team was the second lesson learned. Team members were located throughout the eastern sea board. This required that team members use meeting time effectively. The use of a variety of tools including voice, video, and email allowed team members to convey thoughts and divide tasks. The equal distribution of work and focusing of efforts was of utmost importance. This division of work allowed team

members to work in parallel rather than waiting on others to finish assignments. Work products were then reviewed for quality and consistency with the rest of the project. This communication strategy ensured success.

The importance of pursuing multiple paths was the final lesson learnt during the course of this project. This success demonstrates the effectiveness of the systems engineering approach. The systems engineering life cycle begins by defining the problem domain. This focus eliminates the potential for becoming fixated on a single solution. Following this phase engineers can begin defining solutions to the problem. Defining multiple solutions facilitates trade analysis and optimal solution selection. Through the use of this process the team skillfully navigated the complexities of developing a real world system.

The project team was met with multiple challenges throughout the semester, both technical and business process related. The team defined multiple solutions prior to engaging in development. During the course of the project the initial web scraper solution was deemed inadequate to deliver the desired capabilities. The team was able to refocus their efforts on developing a CSV parsing tool quickly due to earlier design efforts. This alternate solution had been identified early in the solution development stage, facilitating transition from the web scraper to CSV parser. This success was the direct result of following the systems engineering life cycle process. This demonstrates that the process, if properly tailored, can be applied to projects of all sizes.

7.3.3 Acknowledgements

This project would not have been successful without the consistent effort and feedback from the following stakeholders:

- Project Sponsors: Bryan Barfield and Ally Shi
- Loudoun County: Karen Lanham, Dept. of Economic Development and Diana Witek, Dept. of Information Technology
- Faculty: Dr. Barry, Dr. Huang and Dr. Zaidi

8.0 Appendices

8.1 NOVEC Project Schedule

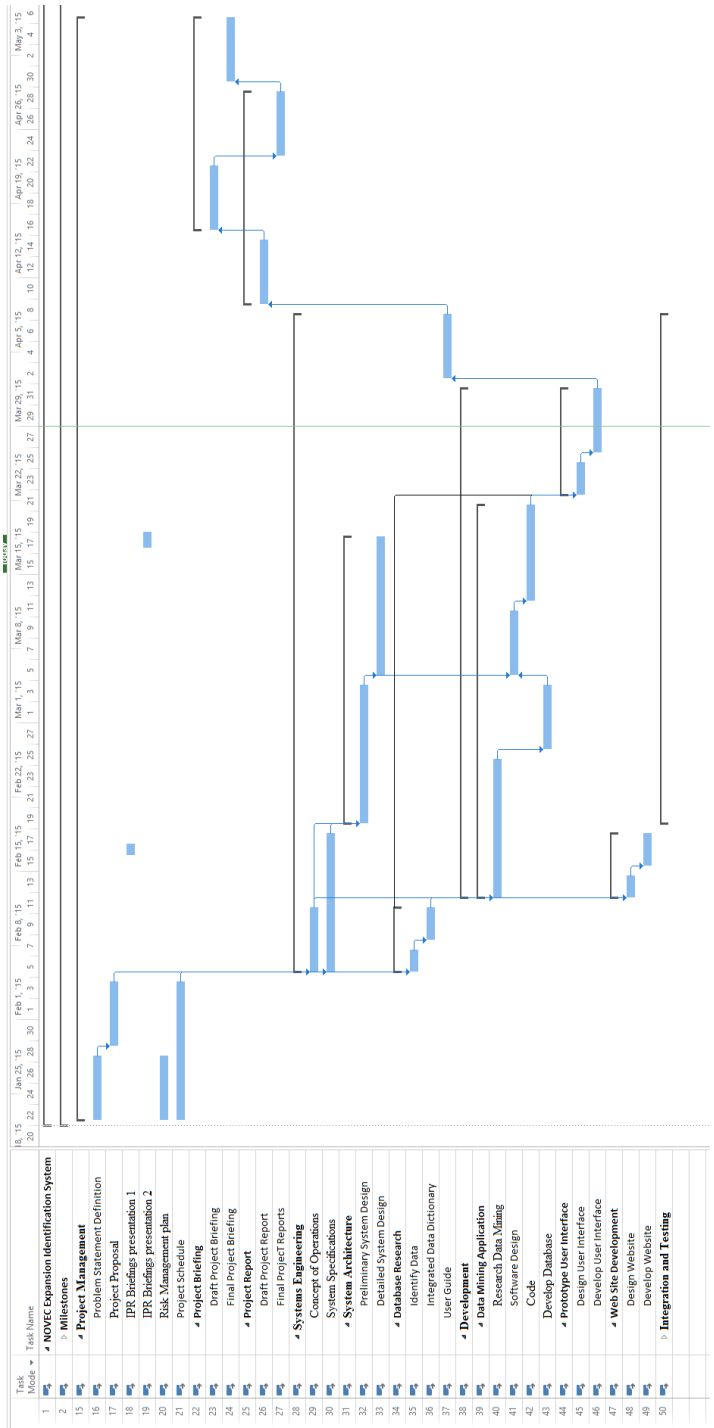


Figure 21: NOVEC Project Schedule

8.2 Integrated Dictionary

Term	Definition
AV	All Viewpoint
CRO	County Records Office
CSV	Comma Separated Values
CV	Capability Viewpoint
DCOM	Distributed Component Object Model
DIV	Data and Information Viewpoint
DOC	Microsoft Word file extension
DOCX	Microsoft Word file extension 2007+
DoD	United States Department of Defense
DoDAF	Department of Defense Architecture Framework
ELP	Enterprise Land Parcels
ENS	Electronic Notification System
ESB	Enterprise Service Bus
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
IP	Internet Protocol
IPV4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
JPEG	Image File Interchange Format
LC	Loudoun County
LCAL	Loudoun County Available Land System
LCPZO	Loudoun County Planning and Zoning Office
LCS	Loudoun County System
LOLA	Loudoun County Online Land Application System
LP	Land Parcels
MySQL	"My Sequel", "My S-Q-L" RDBMS
NCIT	NOVEC Contract IT Support
NEXIS	NOVEC Expansion Identification System
NIS	NOVEC Identification System
NOVA	Northern Virginia
NOVEC	Northern Virginia Electric Cooperative
OOCOR	Office of the Commissioner of the Revenue
OV	Operational Viewpoint
PDF	Portable Document Format
PV	Project Viewpoint
PY	Python File format
RDBMS	Relational Database Management System
RSS	Really Simple Syndication/Rich Site Summary/RDF Site Summary
SHP	shape file format
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture
SRS	System Requirements Specification
SSL	Secure Sockets Layer
StdV	Standard Viewpoint
SV	Systems Viewpoint
SvcV	Services Viewpoint
TCP	Transmission Control Protocol
VSD	Microsoft Visio file extension
VSDX	Microsoft Visio file extension 2013+
WS	Web Service
XLS	Microsoft Excel file extension
XLSX	Microsoft Excel file extension 2007+
XML	eXtensible Markup Language

8.3 Risk Management Plan

8.3.1 Risk Identification

Risk identification is dictated by the categories included in Table 9. Risk Assessment Summary. Each risk as identified in the risk assessment will be weighted based upon probability of occurrence and impact to project scope/schedule/deliverable. Responsibility for risk is shared but task work is dictated in the Risk Responsibilities section. Likewise the risk register is dictated in the Risk Assessment section.

Table 9. Risk Assessment Summary

		Impact				
		1	2	3	4	5
Probability	>0.8		4			
	0.6-0.8		6, 15, 20	5, 19		
	0.4-0.6		18, 21, 22	27	17	
	0.2-0.4		25	14	7, 11, 23, 24	
	0-0.2		16	26	2, 13	1, 3, 8, 9, 12

The risk action plan will be supplemented using several generic responses. The possible responses are:

- Avoidance - Avoidance is the nominal risk case where changes are required and acceptable to the project. A contingency plan is enacted should this risk occur. No mitigation plan accompanies these risks.
- Mitigation - A mitigation plan is determined prior to project start and placed in the risk charter. An additional mitigation plan is put into place due to increased importance of the risk. All risks rated “High” will obtain a risk management plan with both mitigation and contingency components.
- Transference - Inherent to the risk is a source outside of normal operational procedures and a stakeholder other than the team members are essential to executing the contingency plan outlined.
- Deferred - Risk is accepted and changes to the project will be implemented only if feasible and buy in is obtained. Otherwise the changes to be enacted will wait until additional clarification or until next semester’s project.

8.3.2 Risk Responsibilities

All project team members will have access to the risk register dictated in Risk Assessment. The roles and responsibilities for managing risk are outlined as follows:

- Risk Identification and Assessment: All project stakeholders
- Risk Registry: Project Manager
- Risk Response Approval: Team lead with concurrence from sponsor
- Risk Contingency Planning: All team members
- Risk Response Management: All team members
- Risk Reporting: Project Manager

8.3.3 Risk Assessment

Risk plan for project		NOVEC system design			
Assessment team members		Tygue, Brian, Austin			
R#	**Risk**	**prob.**	**Impact**	**Priority**	**Actions**
1	Project sponsor (or major stakeholder) changes agreement with existing requirements	0.1	5	Medium	Response: Avoidance Mitigation: Conduct project sponsor communication as dictated in the communication plan. Ask the correct questions and give suitable information during meetings to obtain buy in and subsequent sign off on established requirements (preferably with actual signature) Contingency: Team will address issues during same meeting/forum or schedule a follow-up clarification meeting with sponsor for guidance. Team will mark changes to all applicable requirements document and will schedule a follow up meeting with sponsor for agreement on either: new requirements or feasibility of requested change. Team will determine impact of possible changes and develop action items offline.
2	Project sponsor (or major stakeholder) changes interpretation of existing requirements/team member misinterprets stakeholder's input to requirements	0.1	4	Medium	Response: Mitigation Mitigation: Conduct project sponsor communication as dictated in the communication plan. Ask the correct questions and give suitable information during meetings to obtain buy in and subsequent sign off on established requirements (preferably with actual signature) Contingency: Team will address issues during same meeting/forum or schedule a follow-up clarification meeting with sponsor for guidance. Team will determine impact of change in terms of documentation modifications and develop action items offline.
3	Project sponsor (or major stakeholder) changes scope of project/team lead needs to request change to scope of project	0.1	5	Medium	Response: Mitigation Mitigation: Conduct project sponsor communication as dictated in the communication plan. Ask the correct questions and give suitable information during meetings to obtain buy in and subsequent sign off for problem statement and measures of success. Communicate all recommended changes to sponsor immediately. Contingency: Team will interpret scope of change and affected documents in terms of action items. Team will schedule a

					meeting/create a forum after action items have been listed and seek buy-in on recommendations. Team will enact changes to document offline with roles and responsibilities decided likewise.
4	Project sponsor does not accept recommendations for project changes	0.9	2	Medium	Response: Avoidance Contingency: Team will re-iterate effort, based on new sponsor input, with same roles and responsibilities established in action items.
5	Project sponsor (or major stakholder) requests additional functionality to deliverables	0.8	3	Medium	Response: Avoidance Contingency: Team will delegate tasks as addition to existing action items. If additional software development is required these tasks will require special interpretation from OR SME and agreement from project sponsor before implementation. Action items will be executed.
6	Project timeline changes or deadline for deliverables changes "moves to the left"	0.8	2	Low	Response: Mitigation Mitigation: Allow schedule to be a living document. Anticipate incoming events (blizzards etc.) and plan accordingly. Contingency: Team will immediately develop and delegate action items as a high priority. The list of affected documents and changes will immediately move up in priority.
7	Delegated team member is not able to complete task as assigned in action items	0.4	4	Medium	Response: Avoidance Contingency: Team will determine if skill set is available/able to be developed to complete task. If not team will develop recommendation to project sponsor for scope changes.
8	Delegated team member becomes indisposed for task work	0.2	5	Medium	Response: Deferred Contingency: Team will reallocate tasks based on action item lists. Team will complete tasks in parallel with new tasks (schedule over-time if needed).
9	Group charter changes	0.2	5	Medium	Response: Transference/New team member Contingency: Team will determine skill set of new team member and re-establish charter. Member will reallocate tasks assigned to former colleague.
10	Data from different locations is not compatible	0.8	3	Medium	Response: Deferred Contingency: Team will coordinate with sponsor/stake holders to limit or filter data. Team will identify necessary changes to all documentation. Recommendations will be made to sponsor based on level of effort. If it is agreed to be infeasible to incorporate changes to system outside of current scope, changes will be deferred to next semester. Otherwise system changes will be executed.
11	Sponsor/Stakeholder does not specify the	0.4	4	Medium	Response: Transference/Sponsor Mitigation: Follow communications plan as dictated in project

	important/relevent information or gives low quality input				charter. Answers to all agenda items should be dictated and recorded in meeting minutes. Should this information be lacking and require project changes execute contingency plan. Contingency: Team will discuss clarification with sponsor/stake holder. If necessary, team will make assumptions and detail assumptions in documentation to be communicated to sponsor.
12	Scope becomes inflated	0.1	5	Medium	Response: Avoidance Contingency: Scope will be adjudicated in next meeting and recommendations will be made to project sponsor for approval of reduced scope.
13	Data gathered is inaccurate	0.2	4	Medium	Response: Mitigation Mitigation: Find format of raw data to be polled from. Find validation source from public record. Contingency: Data validation tool will be developed if data source is found to be inaccurate. (Current data sources: Planning and Zoning department's Loudoun Online Land Applications System, Economic Development Department's Loudoun County Available Land database, Loudoun subdivision approval forms). Communicate discrepancies to county.
14	Sponsor specifies incorrect information	0.4	3	Low	Response: Transference/Sponsor Mitigation: Follow communications plan as dictated in project charter. Answers to all agenda items should be dictated and recorded in meeting minutes. Should this information be incorrect and require project changes execute contingency plan. Contingency: Team will discuss with sponsor/stake holder and obtain clarification. If necessary, team will make assumptions and detail assumptions in documentation.
15	Time required estimates are too low	0.8	2	Low	Response: Avoidance Contingency: Team will adjudicate schedule. Team will meet to develop action items. Team will work overtime to complete tasks as dictated by the action items.
16	Change management system is insufficient for project scope	0.2	2	Low	Response: Mitigation Mitigation: Execute change management system as dictated in the proposal. Google drive is to be used for all versioning. Meeting minutes will be taken for all actions and documents resulting from actions will result in new document versions. Contingency: Current system of google drive and naming documents with version numbers, will be supplemented with dedicated change control personnel, and a change approval board (team members).
17	Activities are missing from scope	0.6	4	Medium	Response: Avoidance Contingency: Scope will be adjudicated in next meeting and recommendations will be made to sponsor for increased scope.

18	Major stakeholder or project sponsor becomes indisposed or disengaged	0.6	2	Low	Response: Mitigation Mitigation: Develop acceptable communication plan. Accept input from all appropriate stakeholders and sign off. Make communication plan a living document. Contingency: Find an alternative stakeholder to receive approval on project changes. (i.e. Sponsor/Professor and vica-versa)
19	Stakholders develop a fundamental disagreement of project scope/intent/requirements	0.8	3	Medium	Response: Avoidance Contingency: Team will adjudicate all disagreeable items and find a compromising position. These recommendations will be provided to both stakeholders preferably in the same forum. Action items will be developed and assigned to execute recommendations.
20	Users/Teammembers/ Stakeholder/Sponsor have unrealistic expectations of project outcome	0.8	2	Low	Response: Avoidance Contingency: Team members will adjudicate scope of project and present recommendations to all appropriate parties. Action items will be developed and assigned in follow-up meeting.
21	Design/deliverables are not fit for purpose	0.6	2	Low	Response: Mitigation Mitigation: Communicate with sponsor on impact of design changes. Keep good change process. Allow design to evolve based upon architectural feedback. Contingency: Effort will be reiterated and changes will be adjudicated. Product changes will be presented to sponsor and all affected stakeholders in next IPR.
22	Design/deliverables lack flexibility	0.6	2	Low	Response: Mitigation Mitigation: Communicate with sponsor on impact of design changes. Keep good change process. Allow design to evolve based upon architectural feedback. Contingency: Effort will be reiterated and changes will be adjudicated. Product changes will be presented to sponsor and all affected stakeholders in next IPR.
23	Data analysis is not in good format	0.3	4	Medium	Response: Mitigation Mitigation: Research possible sources of database. Receive buy in on concept of operations and data dictionary. Contingency: Obtain additional information from website and data sources. Identify process that is not automated (CSV data dump).
24	Failure to acquire data	0.4	4	Medium	Response: Mitigation Mitigation: Make sample/test data to test all components of the system. Develop test plan for contingency (test) data while data is sourced for proof of concept. Contingency: Develop sample/test data.
25	FOIA Application	0.2	2	Low	Response: Avoidance

	needed				Contingency: Fill out FOIA application and communicate need to Loudoun County FOIA representative.
26	Loudoun County does not approve data usage	0.1	3	Low	Response: Mitigation Mitigation: Maintain communication with Loudoun County as a stakeholder. Communicate need and receive buy in from data stewards. Contingency: Cite public use status of data. Develop test data for concept of operations based on format of data in LOLA.
27	Raw data is out of date	0.6	3	Medium	Response: Avoidance Contingency: Implement manual collection process in the form of CSV dump

8.3.4 Risk Management Reporting

The risk management plan is a living document that will allow for input from all stakeholders. As such changes to it will be documented and addressed in all meeting minutes. The risk register will accompany the project proposal and if no changes are identified this version will act as the baseline.

If a risk occurs the risk will be identified in the meeting minutes and appropriate action items will be adjudicated in the same meeting if possible. If additional stakeholders are required a follow up meeting will occur. The reported risk will be recorded as complete when all action items are addressed from the adjudicating meeting members.

Should the encountered risk be unforeseen the risk register will be updated with an appropriate action plan adjudicated in the next meeting as dictated in the communications plan.

8.3.5 Accompanying Processes

The risk management plan will be supplemented by the following processes with rationale:

- The change management process will apply to the risk management plan.
- The meeting minutes process will record all risk responses and action items resulting from encountered risk.
- All risk will be communicated to the sponsor and professor as dictated in the communication plan.

8.4 NEXIS User's Manual

Table of Contents

Software Requirements

Operating System

NEXIS is assumed to be run on a Windows machine. The python scripts should be portable to other operating systems but there has been no testing done for other operating systems.

Database

NEXIS requires access to a MySQL database. MySQL can be obtained from http://dev.mysql.com/downloads/windows/ for windows machines

Python 2.7

NEXIS is built using Python 2.7. Python 2.7 can be downloaded at https://www.python.org/download/releases/2.7/ .

Python 2.7 can be installed as part of *Anaconda* from Continuum Analytics, a "*(c)ompletely free enterprise-ready Python distribution for large-scale data processing, predictive analytics, and scientific computing.*" https://store.continuum.io/cshop/anaconda/ Anaconda includes the design environment Spyder.

NEXIS requires the following packages: datetime, MySQLdb, os, fiona, shapely, pyproj, csv. To install the packages with Anaconda, open an Anaconda Command Prompt and type *conda install <package name>* or *pip install <package name>*. To install using python, go to python27 folder and type *pip install <package name>*.

Sometimes firewalls can prevent installation of packages; you can download the packages directly from http://www.lfd.uci.edu/~gohlke/pythonlibs/ as .whl files. Then type *pip install <whl file name>* from the command prompt in the directory where the whl file is located.

Installation Instructions

Database

To create the NEXIS schema, connect to the database. Run createNexisSchema.sql. The name of the database server must be up to date in nexisDButils.py in the "__init" section.

NEXIS

Create a folder for NEXIS. Place the following source files in this folder: logger.py, lolaUpdate.py, loudounReports.py, nexisDButils.py, nexisInterface.py, records.py. Place the csv record with updates in the same folder. NEXIS will create nexislogs and nexisreports folders in that directory to store logs and reports.

Service Area

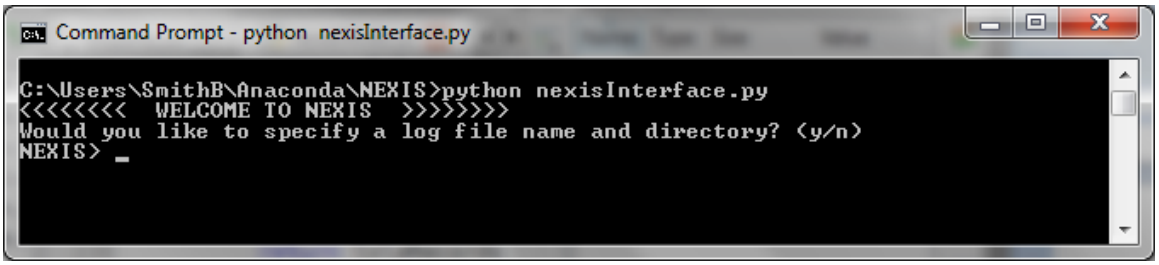
The folder containing the shape file showing NOVEC's service area must be updated in loudounReports.py in the "findRecordsInServiceArea" section of loudounReports.py. Currently, it is assumed the shape file is using EPSG 2283 coordinates. If a different coordinate system is used, this must also be updated in the "findRecordsInServiceArea" section of loudounReports.py.

Using NEXIS

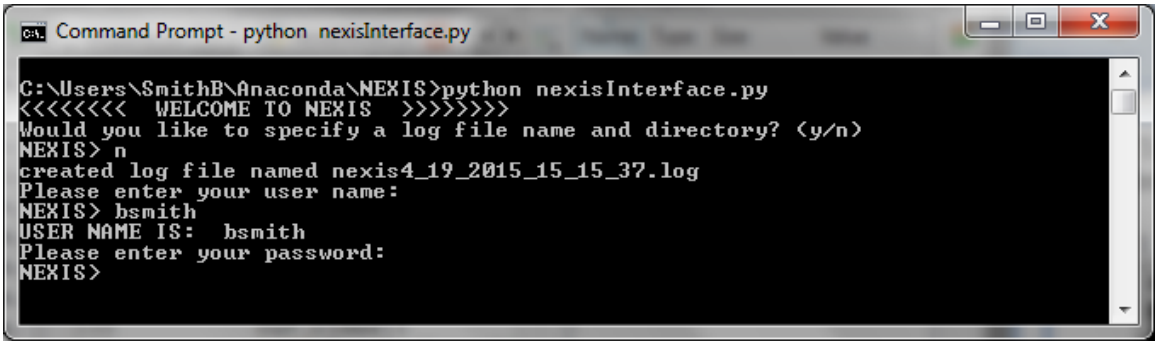
To start NEXIS, open a command prompt and move to the NEXIS folder.

1. At the command line, type *python nexisInterface.py*

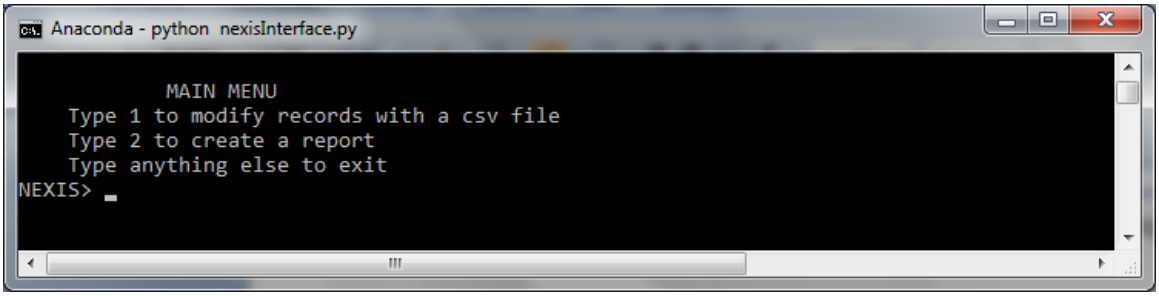
2. The first option is whether to specify the log file name or whether the log file will automatically be generated (recommended). The only acceptable responses are "y" or "n". Other responses will exit NEXIS. If the log file is automatically generated, the log file name will be "nexis" followed by the date and time and ".log"



3. The user will then need to input their database user name and password:

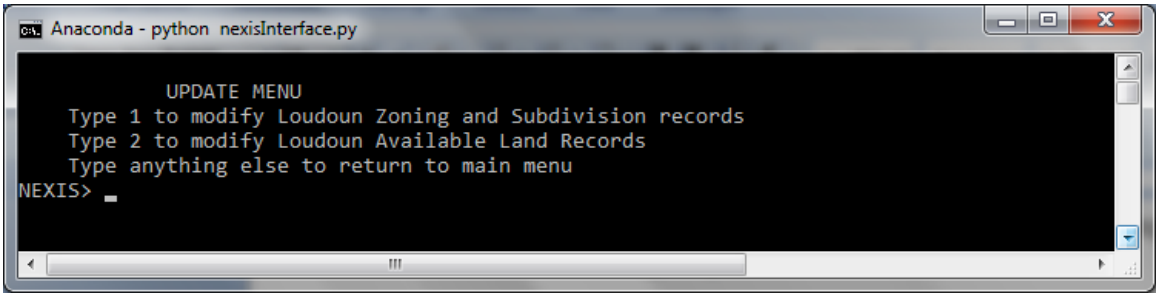


4. The user will then be able to choose between modifying the records and creating a report.

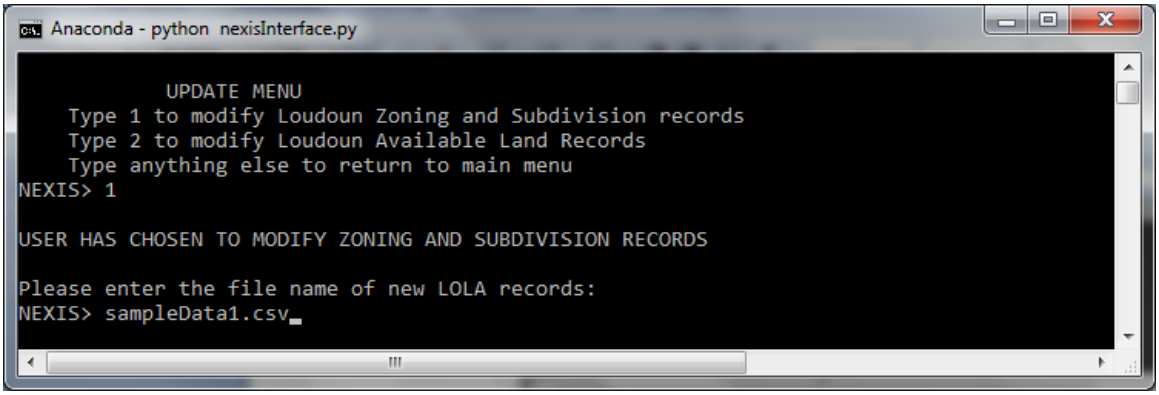


Modifying Records

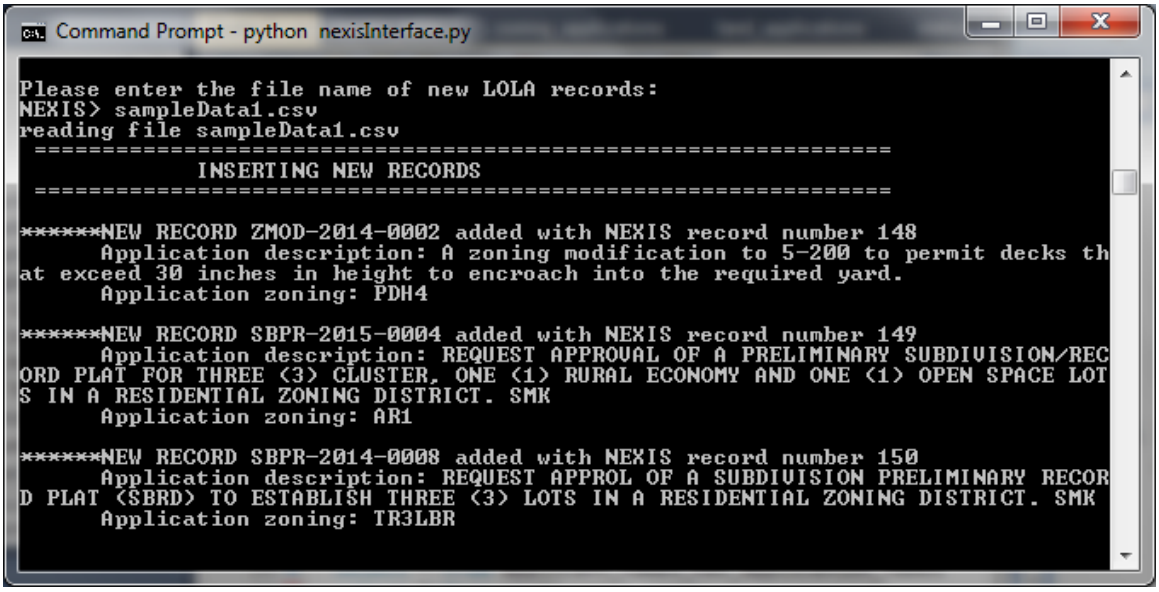
1. To modify the records, choose 1. You will then proceed to the update, menu. Currently only the Loudoun Zoning and Subdivision records can be modified.

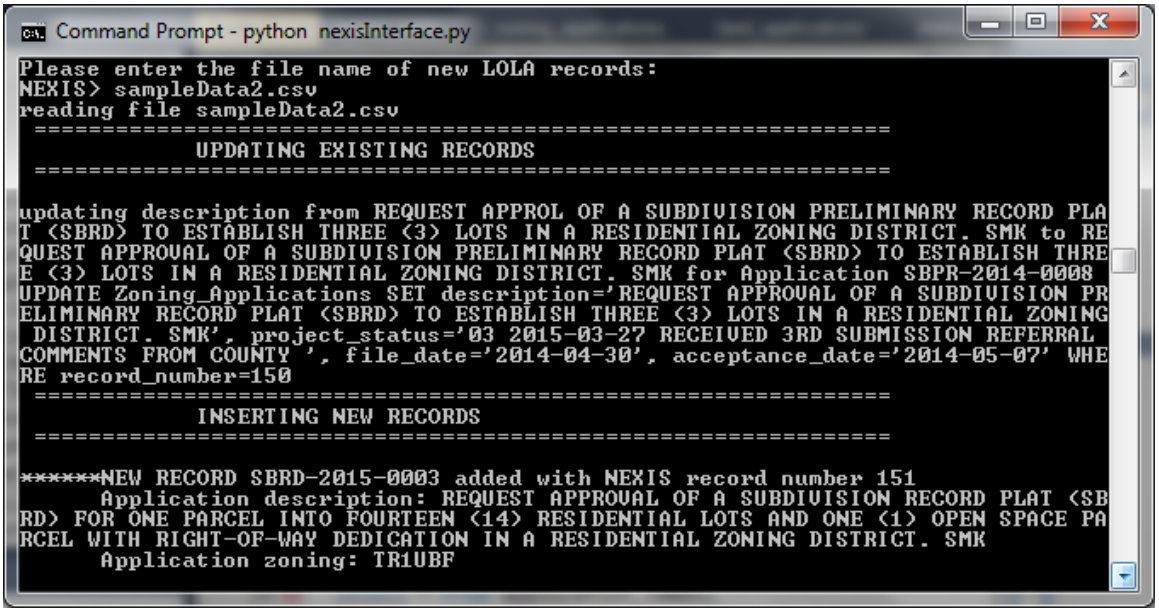


2. Input the file name (or file name and path if the file is not in the same directory as nexisInterface.py) of the csv with modified records.

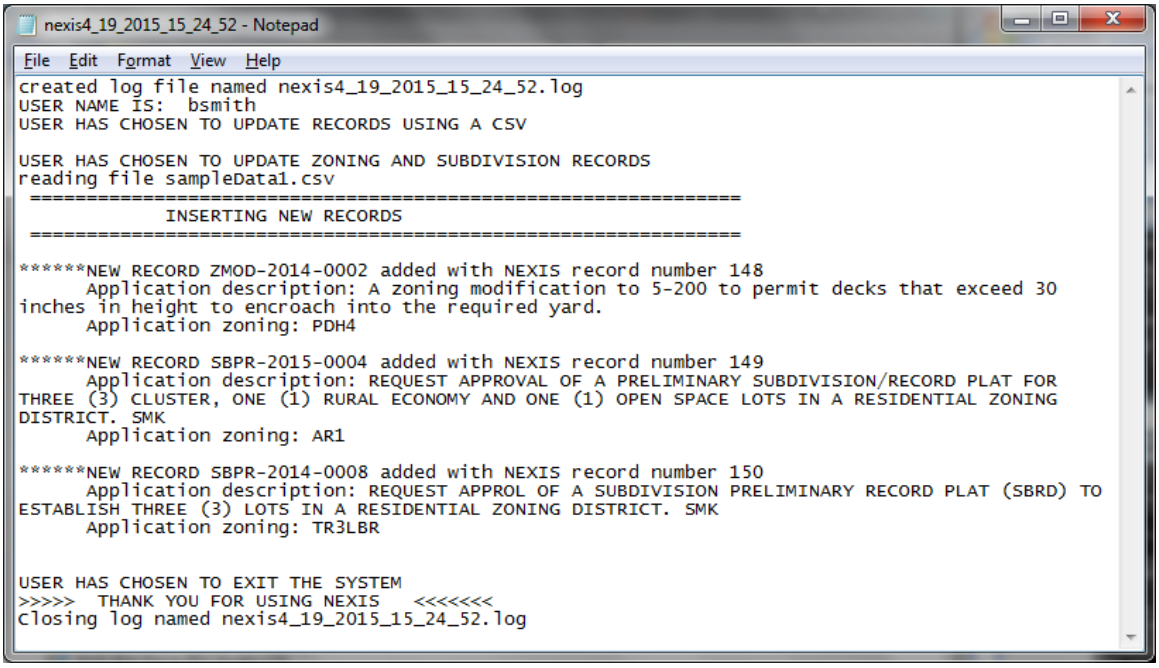


3. The system will search for existing records first and insert new records second. For existing records, it will report where the records are altered. For new records, it will report the Application ID, the NEXIS record number, the application description and the application zoning.

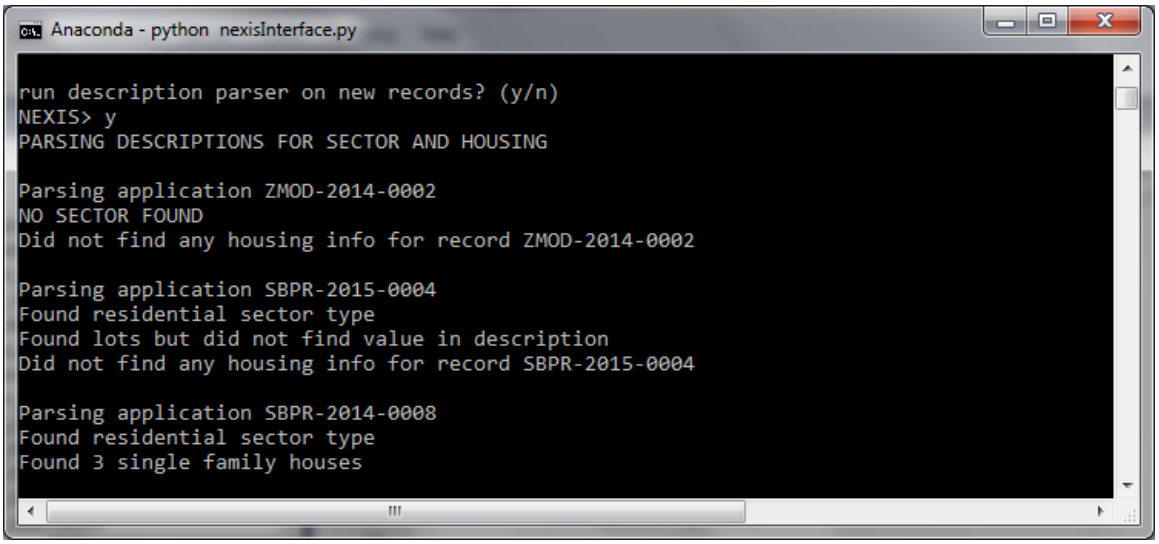




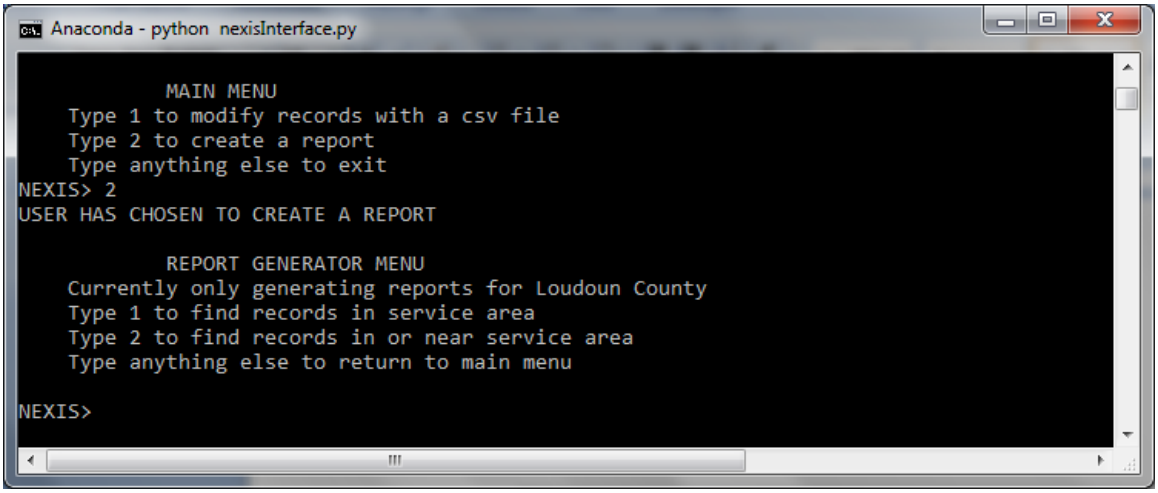
4. The information that is recorded on the screen is also recorded in the log file.



5. After the system completes modifying existing records and inserting new records, the user is given the option to run the description parser on new records. The description parser will attempt to identify the sector for each record and identify any housing units. For details on description parser rules, see descParser.py in Section 5.

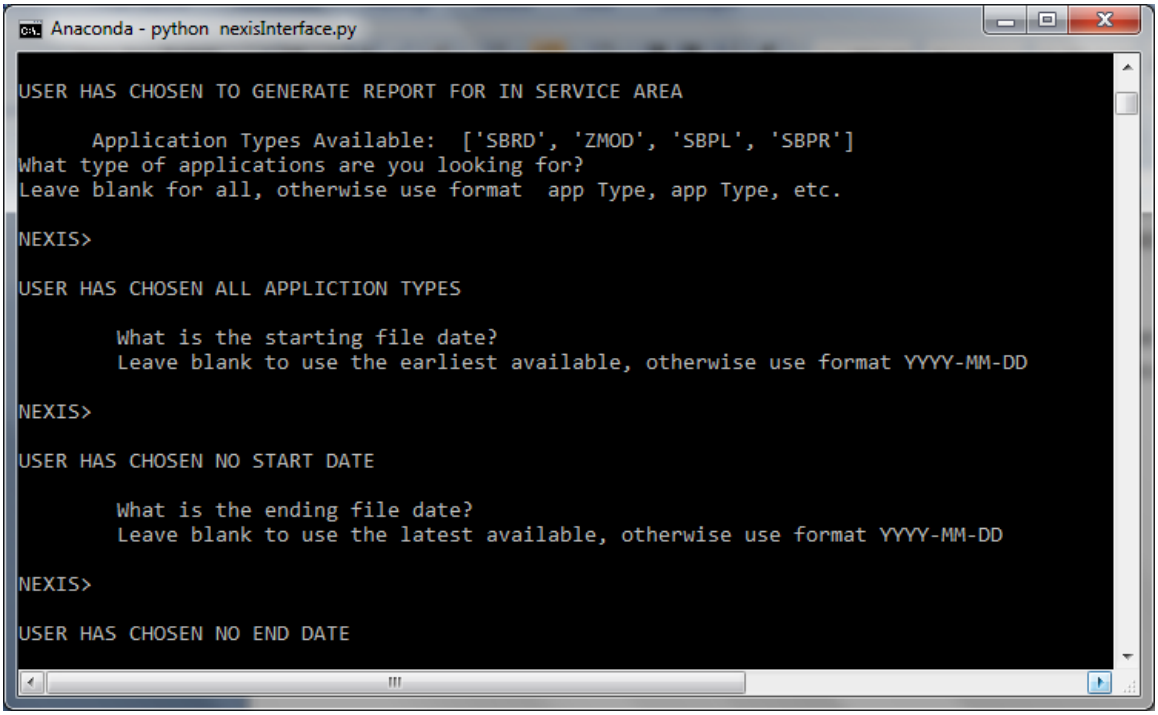


1. To create a report, choose 2 from the main menu.

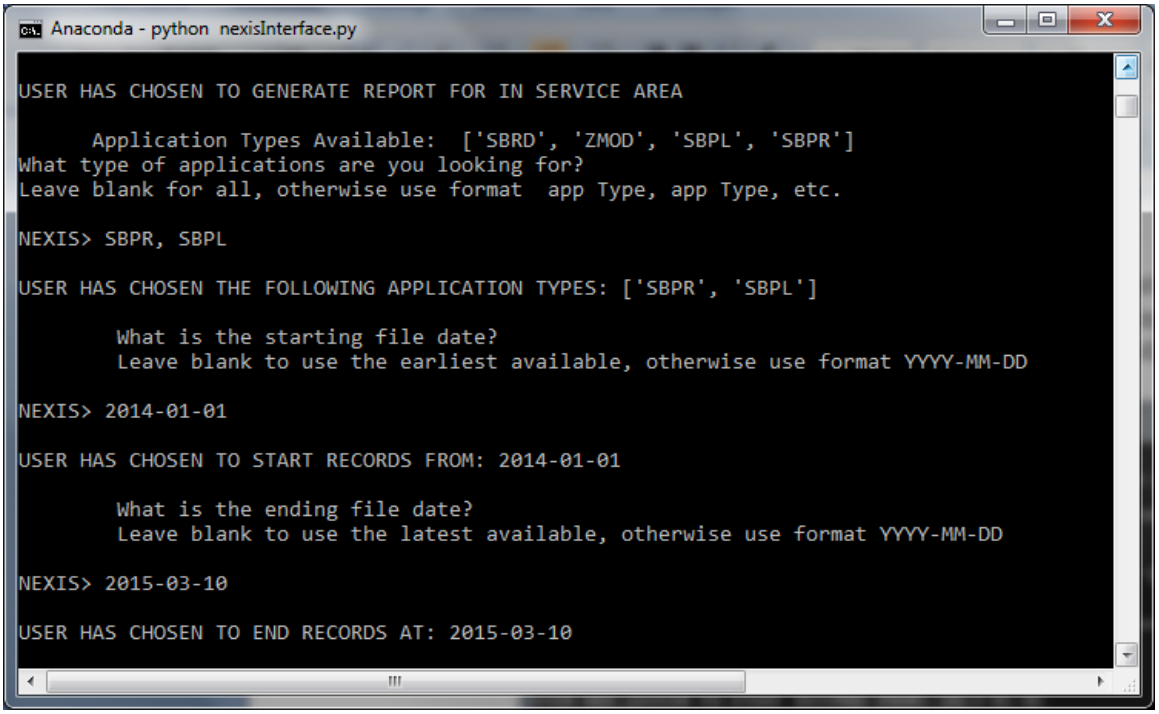


2. The user can choose to determine whether the records are in NOVEC's service area (1) or whether the records are in or near the service area (2).

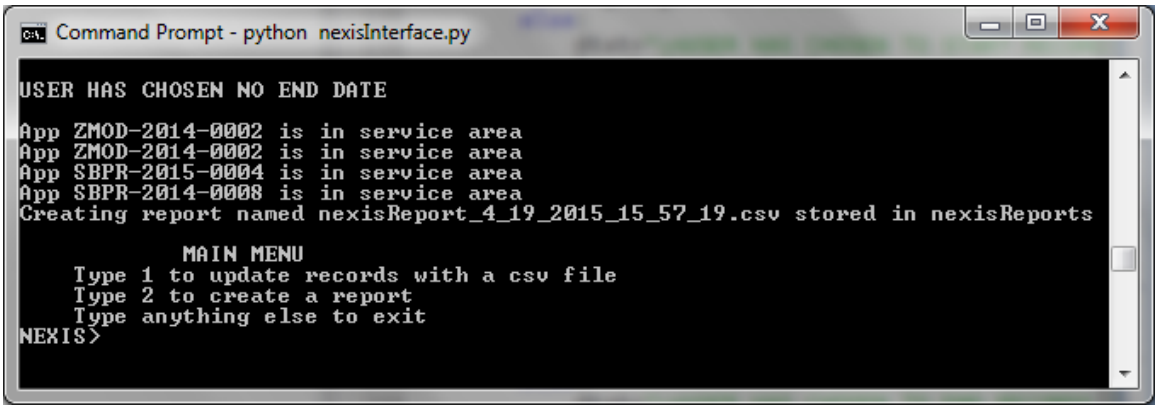
3. Both reports are customizable as to application type and starting file date and ending file date. To search for all, leave that field blank.



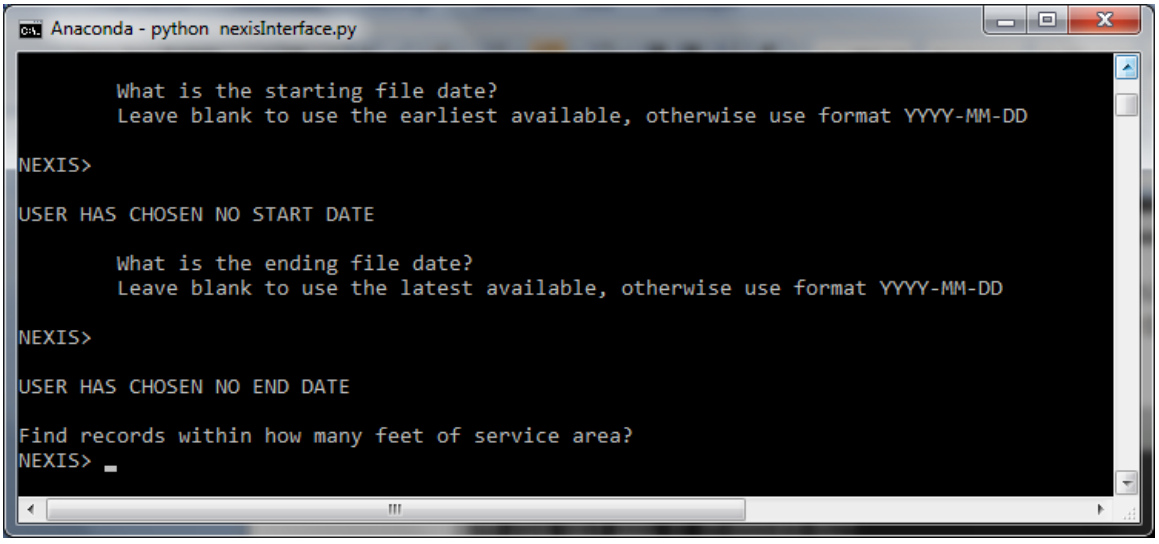
4. To specify application types, input in the form *application type*, *application type*, *application type*. To specify a date, input in the form YYYY-MM-DD (for example, 2015-05-08 for May 8, 2015)



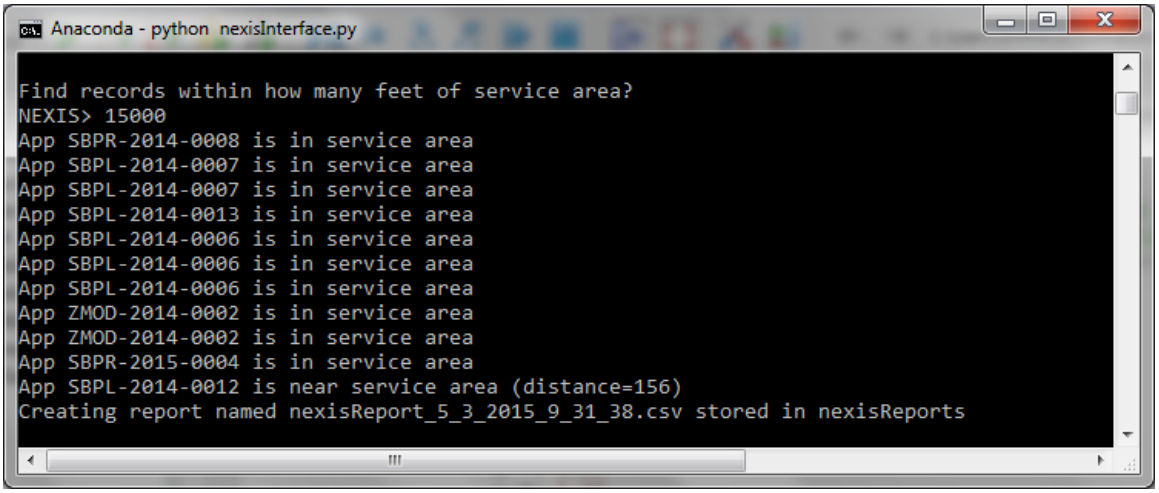
5. For determining which records are in the service area, NEXIS will report to the screen any records that it found and write those records into a report.



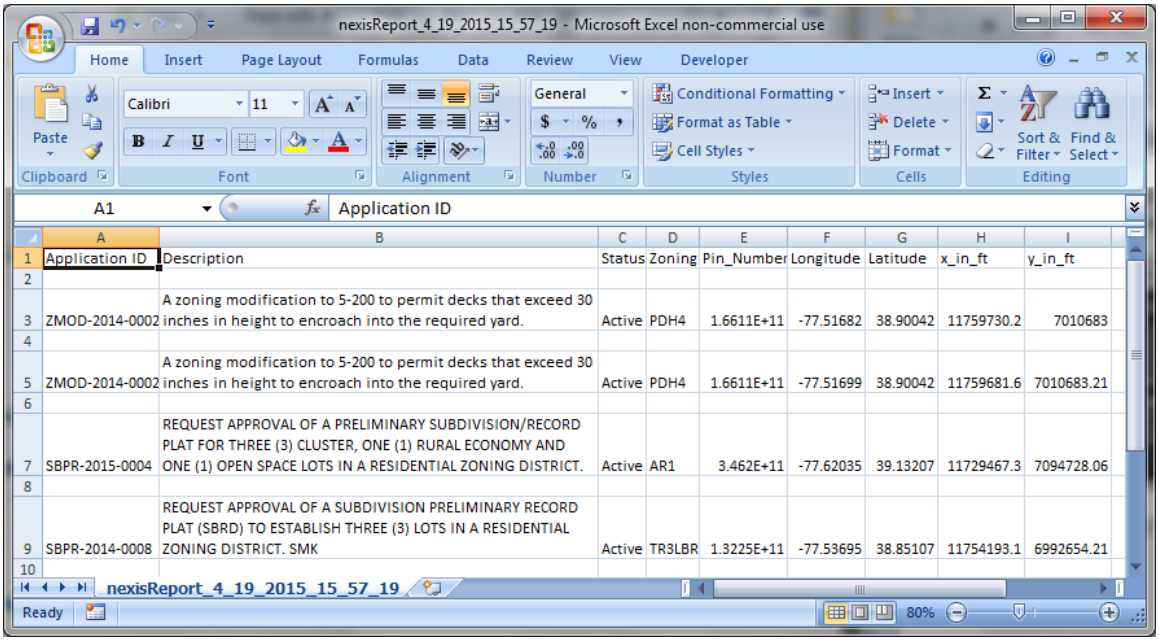
6. For determining which records are in the service area and near the service area, specify the desired proximity (in feet) to the service area border.



7. NEXIS will first determine which records are within the service area. For the remaining records, it will determine whether the records are within the specified proximity threshold. It will write all identified records to a report.

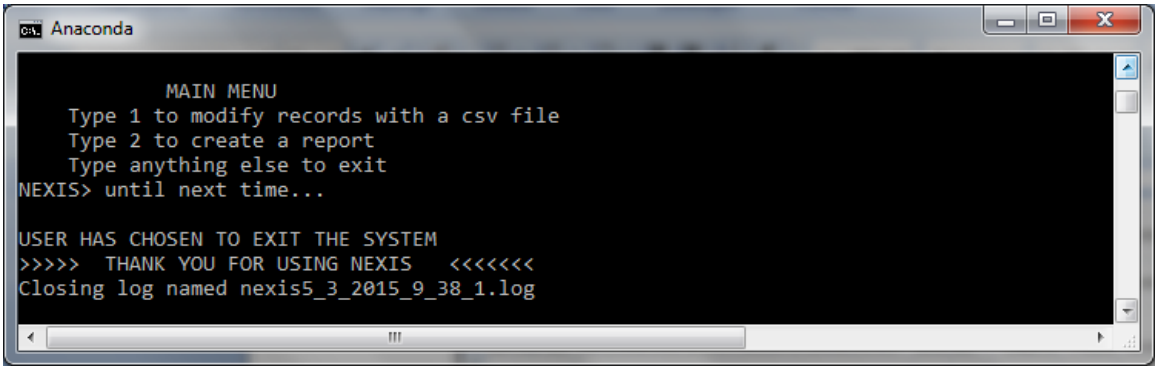


8. The report file will be written in to the *nexisReports* folder in the NEXIS directory. The report will be written to a file named "nexisReports" followed by the date and time and ".csv"



	A	B	C	D	E	F	G	H	I
1	Application ID	Description	Status	Zoning	Pin_Number	Longitude	Latitude	x_in_ft	y_in_ft
2									
3	ZMOD-2014-0002	A zoning modification to 5-200 to permit decks that exceed 30 inches in height to encroach into the required yard.	Active	PDH4	1.6611E+11	-77.51682	38.90042	11759730.2	7010683
4									
5	ZMOD-2014-0002	A zoning modification to 5-200 to permit decks that exceed 30 inches in height to encroach into the required yard.	Active	PDH4	1.6611E+11	-77.51699	38.90042	11759681.6	7010683.21
6									
7	SBPR-2015-0004	REQUEST APPROVAL OF A PRELIMINARY SUBDIVISION/RECORD PLAT FOR THREE (3) CLUSTER, ONE (1) RURAL ECONOMY AND ONE (1) OPEN SPACE LOTS IN A RESIDENTIAL ZONING DISTRICT.	Active	AR1	3.462E+11	-77.62035	39.13207	11729467.3	7094728.06
8									
9	SBPR-2014-0008	REQUEST APPROVAL OF A SUBDIVISION PRELIMINARY RECORD PLAT (SBRD) TO ESTABLISH THREE (3) LOTS IN A RESIDENTIAL ZONING DISTRICT. SMK	Active	TR3LBR	1.3225E+11	-77.53695	38.85107	11754193.1	6992654.21
10									

9. When an action is complete, the user will return to the main menu and can exit by typing anything other than 1 or 2. NEXIS will close the log and remind the user the name of the log file.



Input Formats

Loudoun Zoning and Subdivision records

The Loudoun Zoning and Subdivision records must be in csv format. The information must be in the following order: Application ID, Status, Application Name, Description, Project Status, Applicant Name 1, Applicant Name 2, File Data(DD/MM/YYYY), Acceptance Date(DD/MM/YYYY), Completion Date(DD/MM/YYYY), Existing Zoning, Pin Number, Tax Map, Latitude, Longitude. This file should not include a header row.

Service Area

The folder containing the shape file showing NOVEC's service area must be included to determine reports. To update the service area, please refer to section 2. Installation Instructions.

Using NEXIS Source Code

nexisInterface.py

nexisInterface controls the flow of actions for NEXIS. If new options to update records are added or new reports are generated, the user would need to update these menus to reflect the additional functions

logger.py

logger is the controls for creating, reading and writing to the log file. If the user wanted to change those functions, this is where they would do it.

records.py

records contains the records as objects. If new record types are added, the user can create them here. For fields that may contain multiple records, the user may want to make extra functions. For example, a zoning application may have multiple applicant names, so a list of applicant names is created. The zoning record may have multiple pins with associated tax map, latitude and longitude, so a dictionary is created to store these pins.

nexisDButils.py

nexisDButils contains all interactions with the NEXIS database. This shows examples of selecting various records, updating records and inserting records.

lolaUpdate.py

lolaUpdate updates the Loudoun Zoning and Subdivision records. It checks the new records against the existing records and then inserts new records. Functions included:

updateLolaRecords: controls the main flow through the update, sorts records from csv into existing and new records.

readNewDataFile: reads the csv file and returns lola records.

checkForNewApplicationTypes (or StatusTypes or ZoningTypes): checks the new records type verses the existing types in the database. Currently, new types are added to the database. In the future, if the types are known and static, this can be used to validate incoming data. If a type is not in the existing set, an error can be thrown to signal invalid data.

findGeoUpdates (or StatusUpdates or AppInfoUpdates or AppNameUpdates): checks whether the information is the same for a record in the csv compared to its record in the database.

runDescriptionParser: runs the description parser on new records.

adjustDateFormat: If valid date is received, converts date from mm/dd/yyyy format to yyyy-mm-dd. If no date is received, returns an empty string. If an invalid date is received, an exception is raised.

rowStripper: If a string is received, it trims spaces from the beginning and ending of the string. Otherwise, an empty string is returned.

descriptionCleaner: Trims extra spaces from descriptions.

loudounReports.py

loudounReports creates reports by checking whether records are in NOVEC's service area in Loudoun or near NOVEC's service area. Functions included:

inServiceAreaReportGenerator: controls flow to create reports of records in NOVEC's service area.

inAndNearServiceAreaReportGenerator: controls flow to create reports of records in and near NOVEC's service area.

getRecordsByAppAndDate: asks the user if they would like a subset of records by application type, starting file date and ending file date. It returns all applications according to the user's request.

findRecordsInServiceArea: determines which records are in the service area.

findRecordsInAndNearServiceArea: determines which records are in the service area. It determines for the remaining records which records are near the service area.

writeReport: writes to a csv the records that were sent to it.

validateAppChoice (or Date): checks to make sure the user inputted valid data.

descParser.py

descParser checks an applications description to attempt to determine its sector or the number of housing units.

findSectorType: checks if any of the following words are in the description: residential, commercial, industrial, mixed. If more than one is found, it is considered mixed.

findHousingUnits: attempts to determine from the description the number of housing units for an application. If a specific keyword is found, it checks if there

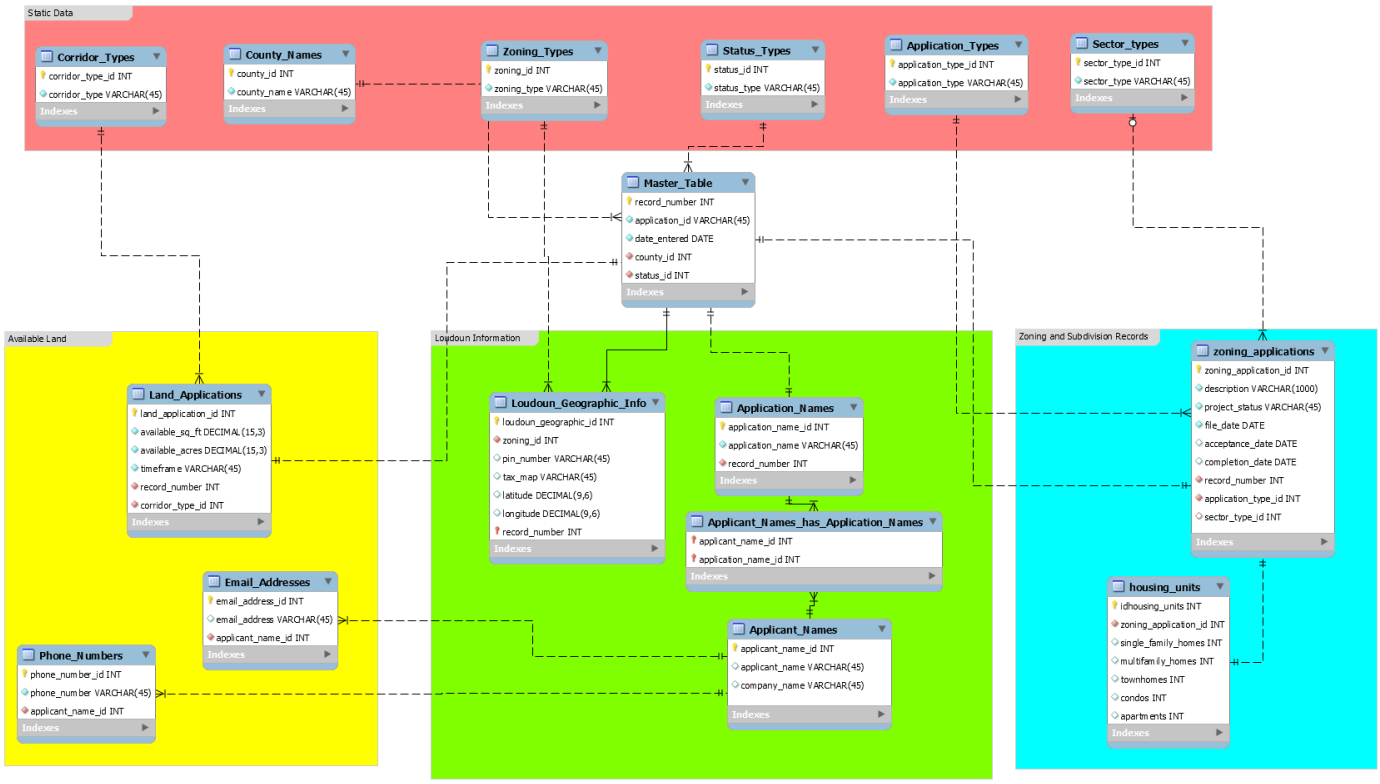
is a number before that keyword (or a number in parenthesis or brackets). The keywords are listed below:

Housing Unit	Keywords
Single Family Homes	sfh, single family home, single family house, lots
Multifamily Homes	mfh, multi family home, multifamily home, multifamily house, multi family house
Townhomes	th, townhome, townhouse, town home, town house
Condominiums	condo, condos, condominium, condominiums
Apartments	apt, apartment

Using NEXIS Database

Querying records

To query records from the database, identify the information required. For tables that are connected, identify the id field that is in common between the two tables.



Deleting records

To delete records from the database, delete records from tables in the following order:

1. applicant_names_has_application_names
2. applicant_names
3. application_names
4. housing_units
5. zoning_applications
6. loudoun_geographic_info
7. master_records

Known Issues

1. If a new application, zoning, or status type is found, the updater inserts that type and highlights that in the log.

2. For lolaUpdate, if the csv has a different applicant name or pin number, it adds the new applicant name or pin number (with tax map, latitude and longitude). It does not remove any existing applicant names or pins.