



Georgia Department of Agriculture

Gary W. Black, Commissioner
19 Martin Luther King Jr. Drive, SW
Atlanta, Georgia 30334-4201
www.agr.georgia.gov

ARMIS – Automated Recall Monitoring Information System Standard Enhancement Projects (SEPs) Georgia Department of Agriculture Food Safety Division

- 1) Rationale and Justification for ARMIS as an SEP
 - a) FSMA mandates a national Integrated Food Safety System (IFSS)
 - b) Updated MFRPS includes surveillance of FOOD-RELATED INCIDENTS (FRI) (5.3.2.2)
 - c) Required documentation is log/database that tracks notification of FRI (5.5.5)
 - d) Only validated data on FRIs are USDA and FDA press releases
- 2) Information on Python
 - a) Supports Object-Oriented Programming and functional programming
 - b) Packages: directories for modules containing functions, classes, variables, etc. built for specified purpose
 - c) Clear, concise syntax
 - d) Often used in web/internet development (Django) & scientific applications (SciPy)
 - e) <https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>
- 3) ARMIS Concept
 - a) Executable written in Python 2.7
 - b) Analyzes RSS feeds and source HTML scraped from press releases
 - c) Inputs data into Excel spreadsheet (.xlsx)
 - d) Uses USDA and FDA RSS feeds (Really Simple Syndication) via feedparser
 - e) Data fields are analyzed and populated using packages:
 - i) Natural Language Toolkit
 - ii) BeautifulSoup
 - iii) Openpyxl
 - f) Graphic User Interface made with Tkinter

Timeline: Fiscal Year 2015-2016

- 1) Develop software in Python for periodically reading RSS XML data to scrape HTML from associated URLs.
 - Research Python documentation for appropriate packages.
 - Write code for obtaining RSS XML data, parsing URLs from RSS entries, and scraping HTML.
 - Use HTML data to recover necessary information
 - Program input function to insert data into appropriate data fields in Excel spreadsheet

- 2) Develop a validated dataset for quality assurance.
 - Continue manual data entry via USDA and FDA food and feed recall press releases into Excel spreadsheet using FY2014-2015 spreadsheet format.

- 3) Create textual analysis components to recognize food products, product information, and other data.
 - Research text analysis packages in Python documentation, such as NLTK and BeautifulSoup
 - Find APIs that may help with text analysis and query.
 - Create functionality to deduce Company, Food Item, Adulterants, etc.

- 4) Develop Graphic User Interface.
 - Research GUI python packages to determine most appropriate format.
 - Use desired package to achieve usable GUI (Tkinter)
 - Debug and QA interface.
 - Research methods of cross-platform executables

- 5) Create validation process and codify SOP.
 - Cross check results of ARMIS run with validated dataset.
 - Use non-concordant data for QA and debugging
 - Develop Standard Operating Procedure (SOP) for ARMIS operation.

Timeline: Fiscal Year 2016-2017

- 1) Develop a methodology for auditing ARMIS sensitivity and fidelity.
 - Explore sampling methods
 - Cross-check with spot manual reads of FDA/USDA press releases.
 - Assess the program for inefficiency and retool, if necessary.

- 2) Expand the capabilities and flexibility of ARMIS
 - Use new and existing directories to predict and provide information with potential partners (Minnesota Department of Agriculture has already agreed to help test it in their system).
 - Expand types of exported data formats (e.g. CSV, .dat, SAS).
 - Evaluate the feasibility of integrating a notification system tied to recall audit/effectiveness checks, wherein the FDA 3177 audit form (or equivalent) would be automatically sent out to inspectors for them to complete upon request.
 - Create new functionality, such as analyzing data in real-time (using either Python, Excel or another statistical package), comparing other states, and custom searching.
 - Increase interoperability of system with other programs (e.g. R, SAS, SPSS, Access).

- 3) Investigate integration with G-Force or other notification systems.
 - Research a method to integrate ARMIS with G-Force. If not possible, determine best alternative method for information dissemination (e.g. Email, Sharepoint).
 - Research Python documentation for implementing notification system with ARMIS.
 - Develop coding to integrate ARMIS with G-Force or other notification system

- 4) Modify SOP, if necessary.
 - Use non-concordant data for QA and debugging
 - Edit Standard Operating Procedure (SOP).

Note: Some aspects of the program have changed since the submission of the abstract (Spring of 2016) and ARMIS' current configuration. The program no longer relies on openFDA API due to it being deprecated and unvalidated.

Background

The Food Safety Modernization Act mandates a national Integrated Food Safety System with the goal of facilitating a safer US food supply. For state-level surveillance, food and feed recall data are manually inputted from their respective USDA and FDA press release webpages. ARMIS is designed to reduce human error and streamline surveillance analysis.

Methods

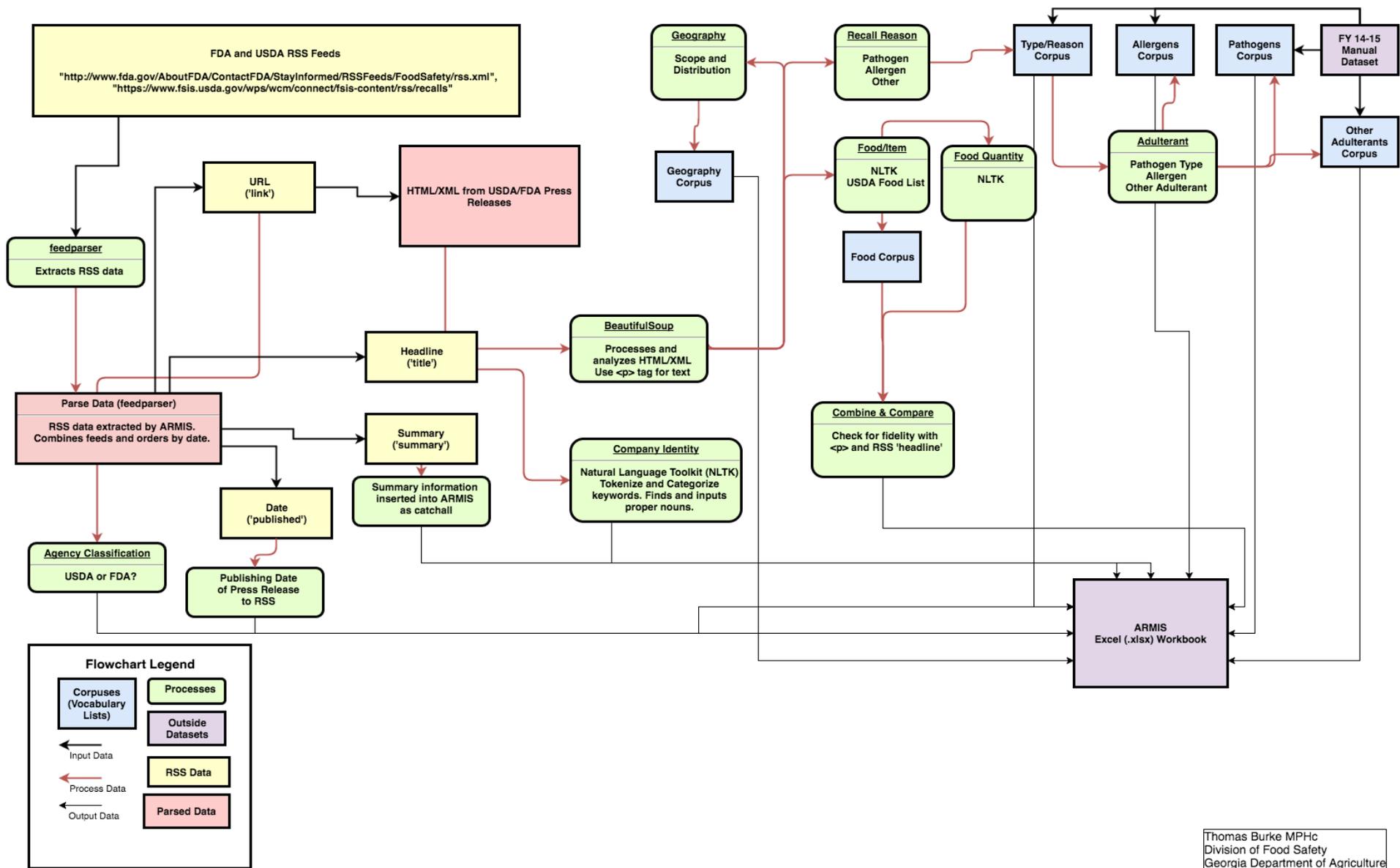
For FDA and USDA recalls, agency press releases are the sole source for validated data. Therefore, to automate data retrieval, ARMIS exploits RSS (Really Simple Syndication) feeds to scrape HTML from each recall press release. Using publicly available datasets and application programming interfaces (APIs), the collected HTML is queried for pertinence to Georgia and specific recall information, such as classification, causation, pathogen or allergen, and distribution. Constructing the program applied web-based data collection techniques from online python documentation and programming texts. Python packages feedparser, openpyxl, and lxml were required for the system's functionality.

Results

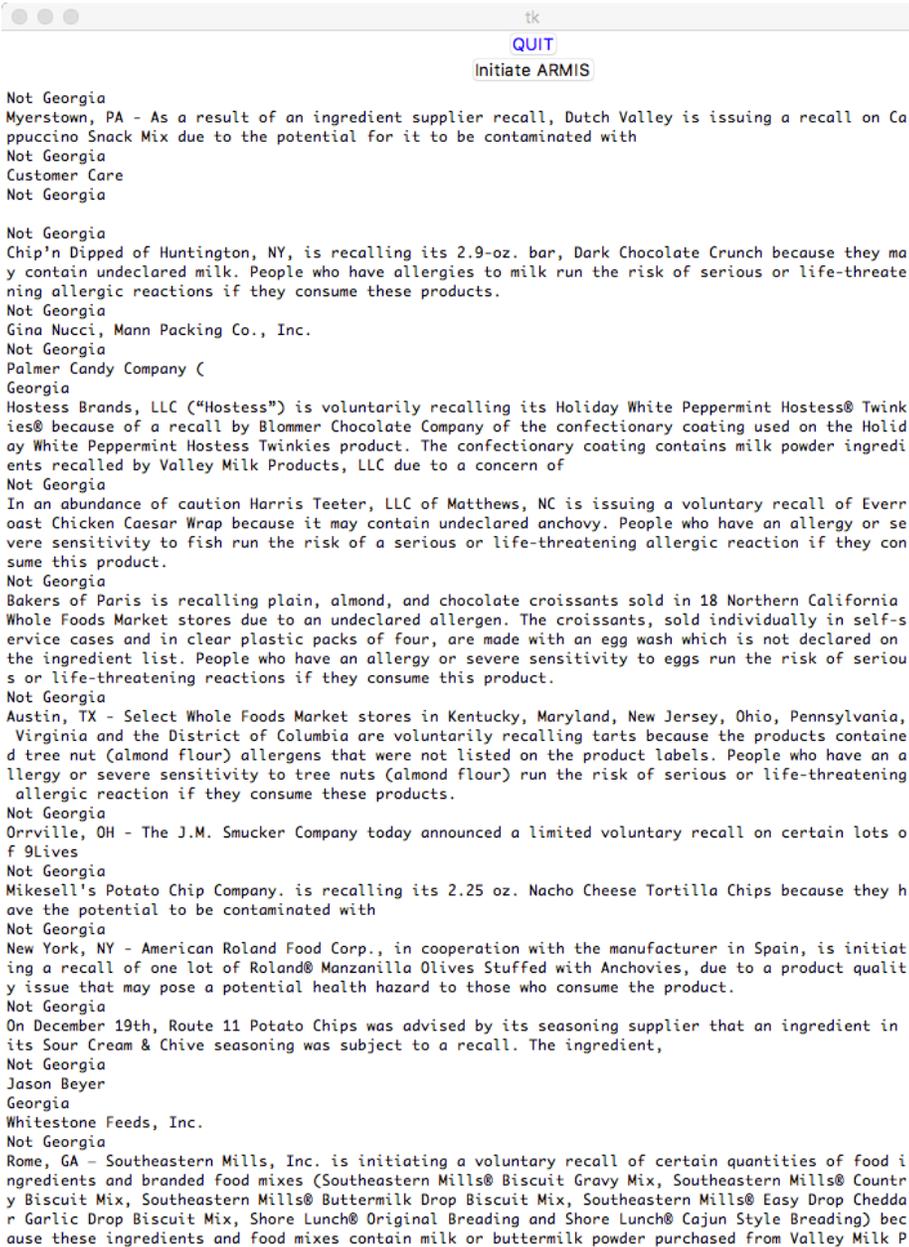
The resultant software program organizes and inputs relevant recall information into a Microsoft Excel spreadsheet. The ARMIS cross-platform graphic user interface design will allow non-programmers to implement the system. Validation methodology is in development to test concordance with current manual data entry.

Implications

We anticipate that ARMIS will be adopted, modified, and extended so all state departments of agriculture may reduce manual data entry and increase data fidelity. Additional phases are planned to extend the surveillance via webcrawling techniques and give ARMIS data mining capabilities.



Thomas Burke MPHc
 Division of Food Safety
 Georgia Department of Agriculture



Top: RSS Feed Raw Output

Left: ARMIS Graphic User Interface. The window is to provide visual confirmation that the program is working.