Weather Triggered Wireless Telemetry

Lightning Talk

Detailed Design

Group: sdmay25-18

Nisha Raj, Alex Chambers, Colin Kempf, Aidan Gull, Adam Fields, Alex Christie

Project Overview

- ARA is an advanced wireless research platform covering Iowa State University, Ames, and nearby rural areas.
- Tasked with creating a system that will recognize and predict when a weather event is occurring.
- This trigger, signals data collection before a given weather event has begun and allows us to continue collecting data until the weather event has passed.
- This weather data will eventually allow researchers to determine how the performance from the ARA framework differs during different weather events.
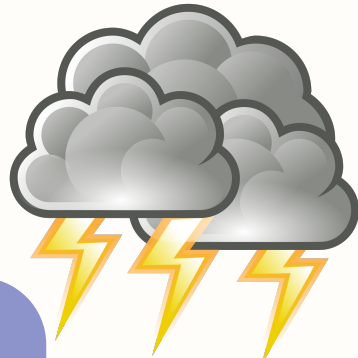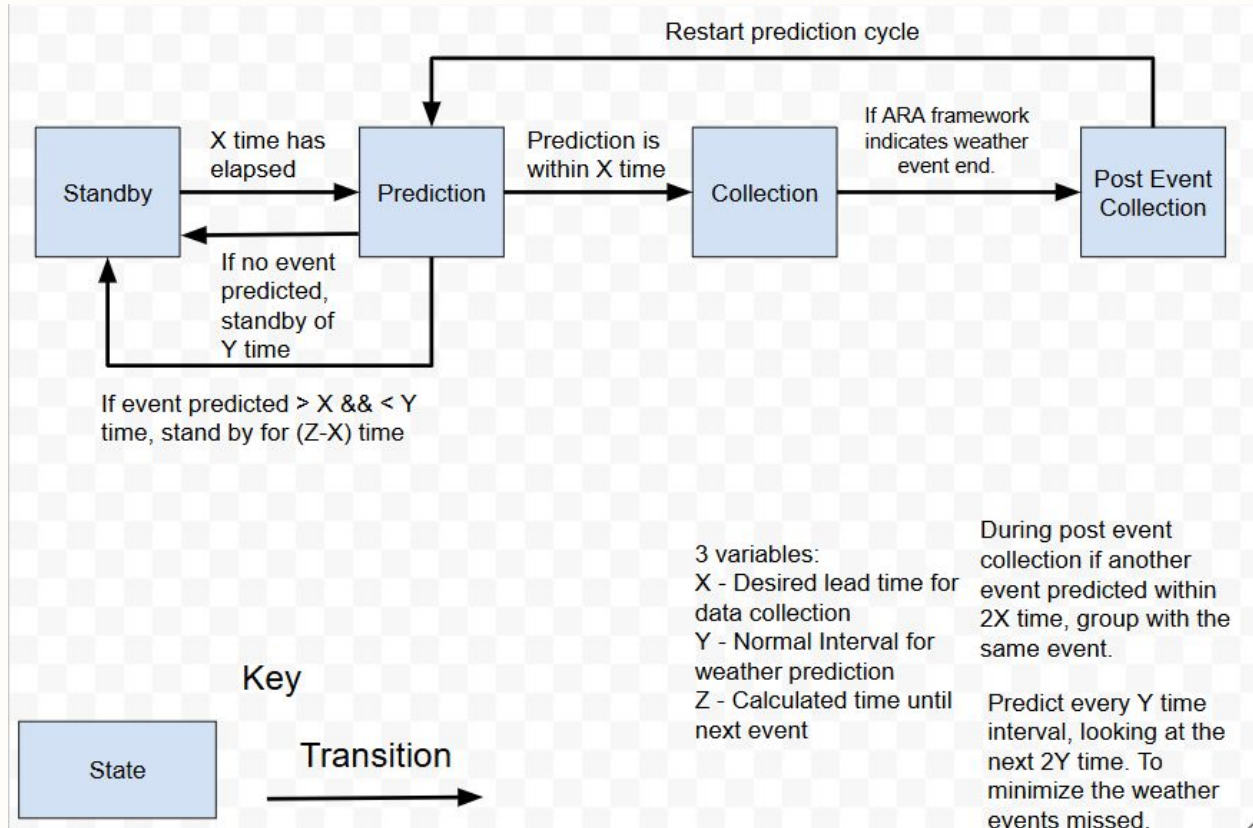

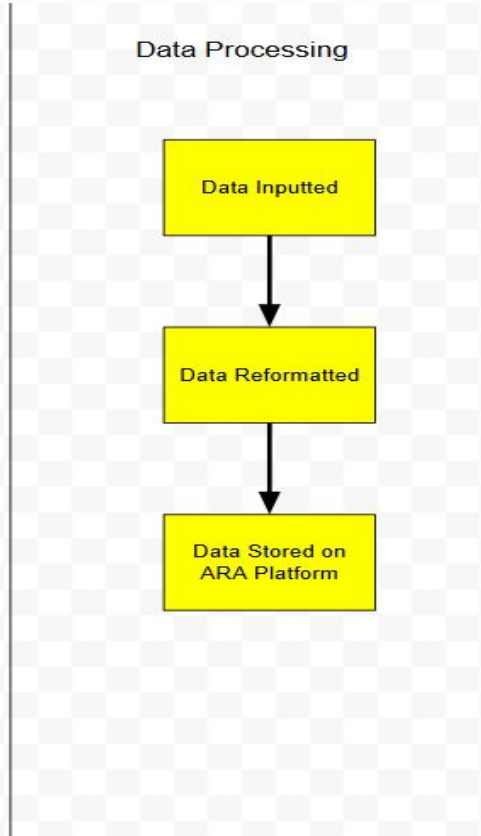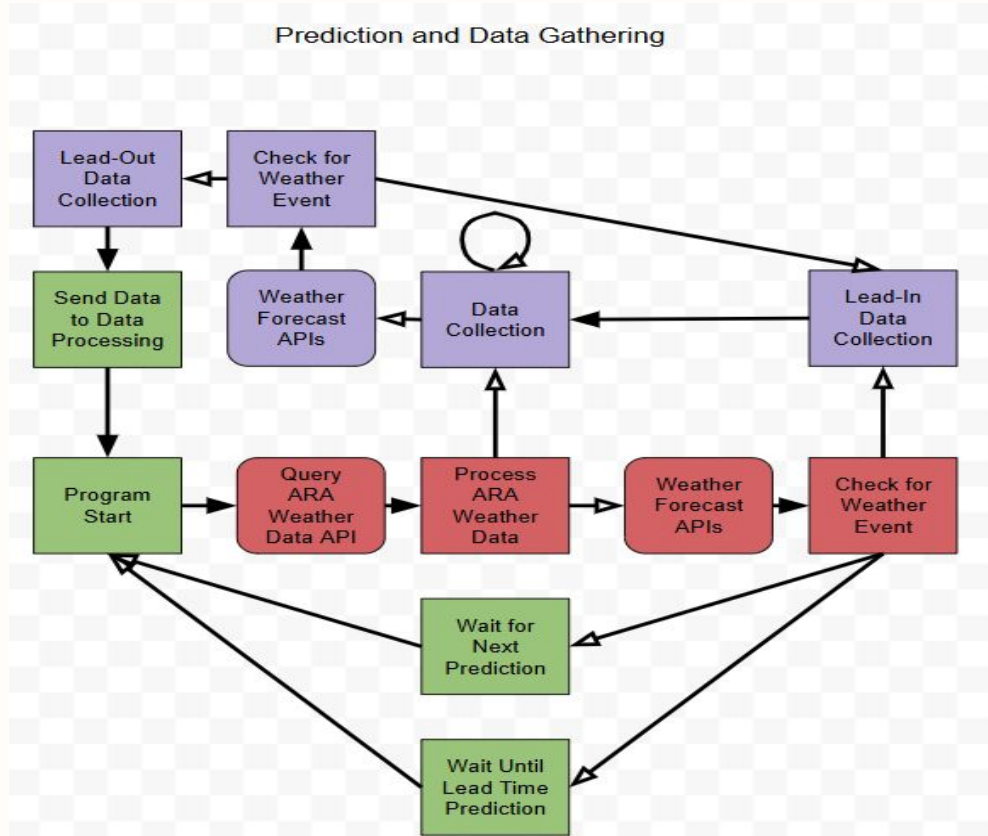
Agronomy Farm

Wilson Hall

Problem Statement

- Want to intelligently collect a wide range of network data during a variety of weather events.

- Use forecast data to predict future weather events to gather data only when weather events we want to record are going to occur.

- Store collected data and allow for user queries to access and format selected data.

State Diagram

Restart prediction cycle

Standby — X time has elapsed → **Prediction** — Prediction is within X time → **Collection** — If ARA framework indicates weather event end. → **Post Event Collection**

If no event predicted, standby of Y time

If event predicted > X && < Y time, stand by for (Z-X) time

Key

State

Transition

3 variables:
X - Desired lead time for data collection
Y - Normal Interval for weather prediction
Z - Calculated time until next event

During post event collection if another event predicted within 2X time, group with the same event.

Predict every Y time interval, looking at the next 2Y time. To minimize the weather events missed.

Predict Weather Diagram

Prediction and Data Gathering

- Lead-Out Data Collection
- Check for Weather Event
- Send Data to Data Processing
- Weather Forecast APIs
- Data Collection
- Lead-In Data Collection
- Program Start
- Query ARA Weather Data API
- Process ARA Weather Data
- Weather Forecast APIs
- Check for Weather Event
- Wait for Next Prediction
- Wait Until Lead Time Prediction

Data Processing

- Data Inputted
- Data Reformatted
- Data Stored on ARA Platform

Functionality

The Diagram on the previous slide breaks the system into 4 logic blocks

- Consolidate Weather Data (Red)
- Consume and Record Active Weather Data (Purple)
- Program Overhead (Green)
- Data Processing (Yellow)

Functionality

Consolidate Weather Data (Red)

- Handles the querying and formatting weather data

- Query ARA API and process data to check for current weather event

- If weather event ongoing move to Consume and Record Active Weather Data (Purple)

  - Else query third party API to find out next weather event will occur

- If next event occurs soon then move to Consume and Record Active Weather Data (Purple)

  - Else move to Program Overhead (Green)

Functionality

Consume and Record Active Weather Data (Purple)

- Specifies records for recognized weather events

- Enter when a weather event is currently ongoing or enter just

  before a weather event (lead-in time)

- Keeps recording data if continued weather events occurring

  - If no more events then will enter lead-out data collection

Functionality

Program Overhead (Green)

- Area of entry to start program and beginning consolidating weather data phase

- Once this block is reached needs to wait for program to start again

  - Finished collecting weather data and need to predict when next event is

  - No predicted weather events within given interval and program should wait that interval to predict again

  - Predicted weather is within given interval and program needs to start predicting over smaller intervals

Functionality

Data Processing (Yellow)

- Handles output data and formats in query format

- Once event is finished data is formatted and stored in ARA's

  platform database

- Dat can then be queries by users from ARA platform to be used

  for visualizations and analysis

Technology Considerations

- Third-party Forecast APIs
  - Strength: Are pre existing, professional resources, that allow us to easily get the data we need.
  - Weakness: Must find free, public, APIs. The data collected from APIs will have a range of accuracy.
  - Trade-offs: We have no control over the code, but we get a much wider range of data for prediction than we could otherwise.
  - Solutions: Use multiple APIs to get multiple opinions on predicted weather events.

Technology Considerations

- ARA Framework & Servers
  - Strength: Client owned and created, with through documentation and resources.
  - Weakness: Must fit within the bounds of the existing ARA framework, both for gathering live data and storing data on their servers.
  - Trade-offs: We get inside access to the ARA backend, but are limited by how much storage space we have. Our program must also fit in with the existing ARA systems without causing problems.

Areas of Concern & Development

Concerns:
- Determining what lead-in time metrics will work best for our prediction calculations?
- What features from the Forecast API gathered data will best allow us to make accurate predictions?

Developments:
- Accounts for false positives and false negatives based on predicted weather
- Ensures that weather events that occur close together in time are gathered together rather than being labeled as two separate events

Conclusion

We have identified:
- Initial state diagram of the project
- Predict weather diagrams
- Distinct technologies being used in design
- Design requirements to meet user needs
- Specific design concerns