

# **Use of Radio Frequency Identification Technology for Seasonal Load Restriction Timing**

**Final Report**  
February 14, 2017

Project Principal Investigator: Paul Fortier, Professor  
University of Massachusetts Dartmouth  
285 Old Westport Road  
North Dartmouth, Massachusetts 02747-2300

USDOT/OST-R Program Manager: Caesar Singh  
OST - Office of the Assistant Secretary for Research and Technology U.S. Department of  
Transportation 1200 New Jersey Avenue, SE, E33-467 Washington, DC 20590  
Contract Number: OASRTRS-14-H-UMDA

**NOTICE:** In May 2020, National Transportation Library staff discovered this submitted report did not have a descriptive title (see page 2). We contacted OST-R Program Manager Caesar Singh, who supplied the title “Use of Radio Frequency Identification Technology for Seasonal Load Restriction Timing.” NTL staff then added this cover sheet to original report. No report content was altered.

2020-05-05

## **Final Report**

Project Principal Investigator: Paul Fortier, Professor  
University of Massachusetts Dartmouth  
285 Old Westport Road  
North Dartmouth, Massachusetts 02747-2300

USDOT/OST-R Program Manager: Caesar Singh  
OST - Office of the Assistant Secretary for Research and Technology  
U.S. Department of Transportation  
1200 New Jersey Avenue, SE, E33-467  
Washington, DC 20590



## **TABLE OF CONTENTS**

<b>Glossary</b>	<b>1</b>
<b>Executive Summary</b>	<b>2</b>
<b>Chapter 1 - Background</b>	<b>3</b>
<b>Chapter 2 - Development of RFID sensors and supporting infrastructure</b>	<b>5</b>
<b>Chapter 3 - Test site identification and deployment of sensor systems</b>	<b>10</b>
<b>Chapter 4 - Decision support system development</b>	<b>12</b>
<b>Chapter 5 - Evaluation of SLR forecasting systems</b>	<b>21</b>
<b>Chapter 6 - Commercialization plan</b>	<b>25</b>
<b>Chapter 7 - Conclusions</b>	<b>27</b>
<b>References</b>	<b>29</b>

DISCLAIMER: The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the USDOT/OST-R, or any State or other entity. USDOT/OST-R does not endorse any third party products or services that may be included in this publication or associated materials.

## **Glossary of Terms**

**API – Applications Programming Interface**

**CITS - Computing and Information Technology Services**

**COTS – Commercial Off The Shelf**

**CSS – Cascading Style Sheets**

**CTI – Cumulative Thawing Index**

**DSS – Decision Support System**

**EICM –Enhanced Integrated Climate Model**

**EPC – Electronic Product Code**

**GEN 2 – Generation 2 protocol**

**HTML – HyperText Markup Language**

**IOT – Internet Of Things**

**MEPDG – Mechanistic Empirical, Pavement Design Guide**

**PM – Program Manager**

**PYTHON - Python's name is derived from the television series *Monty Python's Flying Circus***

**RA – Research Assistant**

**RFID – Radio Frequency Identification, technology for**

**SLR – Seasonal Load Restriction**

**SLRI – Seasonal Load Restriction Interpolator**

**SWE – Sensor Web Enablement standard**

**TAC – technical Advisory Committee**

**UMD – University of Massachusetts Dartmouth**

**USDA – United States Department of Agriculture**

## **Executive Summary**

During the duration of the grant, all proposed task areas of the grant were successfully completed. The technical advisory committee provided the UMD team with leads for commercialization partners and with continual feedback of products usability and improvements. The TAC was instrumental in defining a follow-on grant that helped in moving the initial concepts and prototypes towards commercial products. The TAC has also been instrumental in assisting the UMD Team in publicizing our technology and products to potential clients and investors.

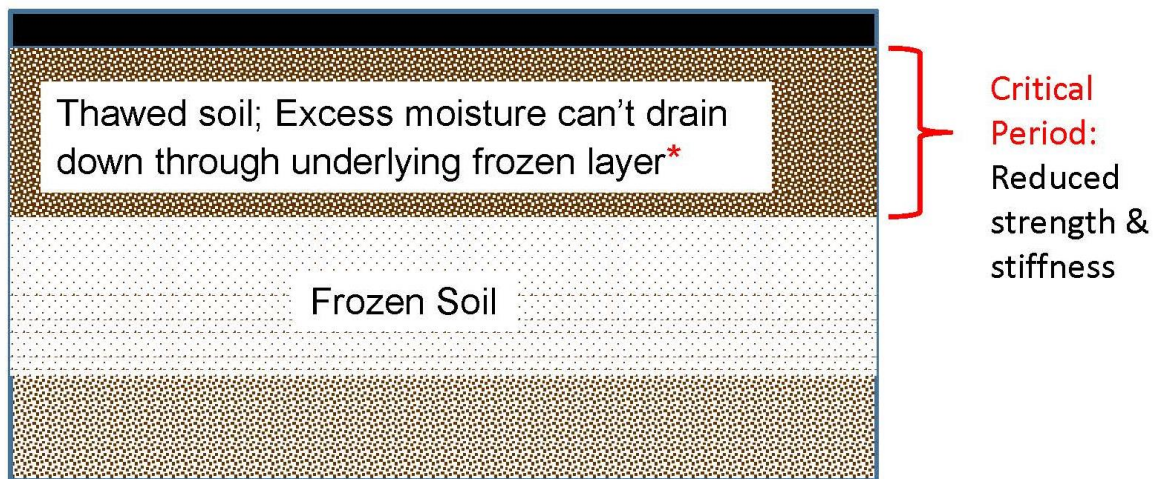
Sensor and communication node design work for the grant were completed last year prior to the request for a no-cost extension. Since then, based upon the needs and feedback from the US Forest Service professional staff additional functionality has been added to the sensor suite as well as to the DSS. An improvement to the sensor head node supports placement of temperature sensors anywhere within the first 18 inches from the roadway surface. The goal of the sensor head node improvement was to provide greater flexibility in sensing near surface conditions. In addition during this extension year improved techniques and methods for sensor node installations were developed as well as adoption of a lower costing communications scheme. Sensors have been installed in six sites with plans for more as new sources of funding are realized.

The UMD SLR DSS system design is complete. Further refinement of the DSS system will depend upon state DOT feedback, industrial customer feedback and commercialization partner's feedback as well as available funds. Project dissemination, outreach and commercialization have been a challenge as the UMD Team worked on marketing and business plan development as well as extending the project through a US Forest Service grant. Commercialization activities continue through negotiations with potential investors and development of further grant and product funding sources.

The study completed a one year no-cost extension to the project that support improved commercialization planning and execution. In a move towards commercialization the UMD Team is in communication with additional state DOT's, potential industrial customers and funding sources to plan next steps. The extended grant time supported development of market and business plans.

## Chapter 1 – Background

Several million miles of secondary paved and unpaved roads in the United States lie in seasonal frost areas. In frost regions, freezing occurs from the surface downward, drawing moisture towards the freezing front where ice lenses form. When roadway layers begin to thaw, underlying layers and composite material are left in an undrained, unconsolidated condition, which are structurally highly susceptible to damage (Figure 1).



**Figure 1:** Roadway Frost – Thaw Concept

Many roads in seasonal frost areas are highly susceptible to damage during the spring thaw period (Figure 2). To understand roadway structural conditions requires knowledge concerning subsurface temperature and moisture. Roadway state information, can be used to develop forecasting models to support transportation infrastructure management. Acquiring roadway surface and subsurface information in real-time is costly, difficult and in some cases, impossible given currently available technologies.

Roadway management policies such as seasonal (or spring) load restriction (SLR) limiting loads of heavy trucks during the spring thaw period are efforts aimed at minimizing costly roadway damage. SLRs, however, pose an economic hardship to truckers responsible for the movement of goods and services over secondary roads. Roadway restrictions may cause trucks to take costly detours requiring additional driving time while hauling lighter loads resulting in more trips. The challenge is to protect the transportation infrastructure and minimize roadway maintenance costs, but also allow commerce to flow as unrestricted as possible during spring thaw and roadway strength recovery periods, typically lasting 6-8 weeks.

Present methods for imposing Seasonal Load Restrictions (SLR)'s are not real-time nor data driven. Most rely on using cumulative thawing index [1] based on computed degree day measurement. Additional studies [2, 3, 4] looked to reduce the time SLR is in place using manual collection methods and models.

State DOT's have performed studies looking at methods to remove SLR's in a timely fashion [5, 6, 7] using measured data, though methods are still not real-time, are costly and mostly manual.



**Figure 2:** Roadway damage due to frost – thaw conditions and uncontrolled use

This project report provides a descriptive overview of the architecture and design of wireless underground radio frequency smart sensors, data collection and Internet of Things (IoT) [8] transmission system and an SLR decision support system. The roadway management system is being tested on the University of Massachusetts Dartmouth's ring road and has been deployed to additional sites on state and federal managed roadways in northern Maine, New Hampshire, Pennsylvania, and Oregon. Additional sites are being defined through interaction with our federal, state, and local DOT partners as well as through commercialization partners such as TrueRoads and TranSense IoT.

Project activities supporting all of the technical and managerial milestones defined in the UMD Tesam's agreement and one year extension have been completed. The results from the ten tasks defined in the original and extended grant will be described below.

## **Chapter 2 – Development of RFID sensors and supporting infrastructure**

Before development of the new RFID technologies could move forward the initial task was to determine if the existing wired sensor systems could be migrated to the new technology easily. It was determined early that this would not be likely as the existing sites were in remote areas where getting clean line of sight for proposed backhaul RF networks was not easily attainable. Based upon these findings it was decided that the UMD Team would attempt to keep the existing wired sites operational as long as possible during the new systems development and during the duration of the grant.

Hard wired sensors at all project phase one sites have failed at one point or another, resulting in loss of data. The existing New Hampshire and Maine sites were visited by our subcontractor Frost Associates to diagnose problems. It was determined that damage was too extensive at the Maine site to bring it back on line. The New Hampshire site hardware has still been causing problems, though the Team have been able to bring it back on line in support of this year's frost-thaw cycle. The project will attempt to find possible replacement funding mechanisms to install the new technology at these phase one sites going forward.

### **2.1 Develop RFID sensors and supporting infrastructure**

All three sub-tasks related to sensor and infrastructure developments were completed successfully and are described in the following text. The first sub Task focused on RFID sensor development. Sensor design, testing, fabrication and installations are completed.

The project's final RF sensor node design (Figure 3) are constructed using three custom designed printed circuit boards (PCB). One PCB is used to read and manage sensor data access, a second PCB is used to manage power generation and a third PCB is designed to manage system interactions.

The sensor node operates at 900 MHz and provide a data rate of 200 kbps. An off the shelf wireless component was chosen to simplify the design and meet project milestones. The communications component is connected to an embedded controller which is responsible for interacting with the sensors to extract measurements, convert raw data to information and package for transmission to the external reader on demand. The controller has 32 MB of RAM and 32 KB ROM used for storing data and firmware implementations. The last element is the power management element that uses a power harvesting unit to maintain battery power to extend operations. The device also supports low power operations between sensor read intervals through a low power operational mode. The antenna design was optimized for low power operations within the noisy environment of the embedded roadway environment [15].

The second sub task completed design, implementation, test and installation of the sensor reader (interrogator), control processor, data and program storage, power management, recharging and RF communications subsystems (Figure 4).



The design, build and testing has been completed and installed at multiple sensor sites. Each node consists of a solar power generation and storage subsystem, XBee radio and antenna (either Omni or directional depending on topography needs). The reader can be operated in a command mode or data transmit mode. In command mode the reader can reconfigure the read rates and upload rates for the subsurface sensors as well as adjust sleep modes and low power intervals.

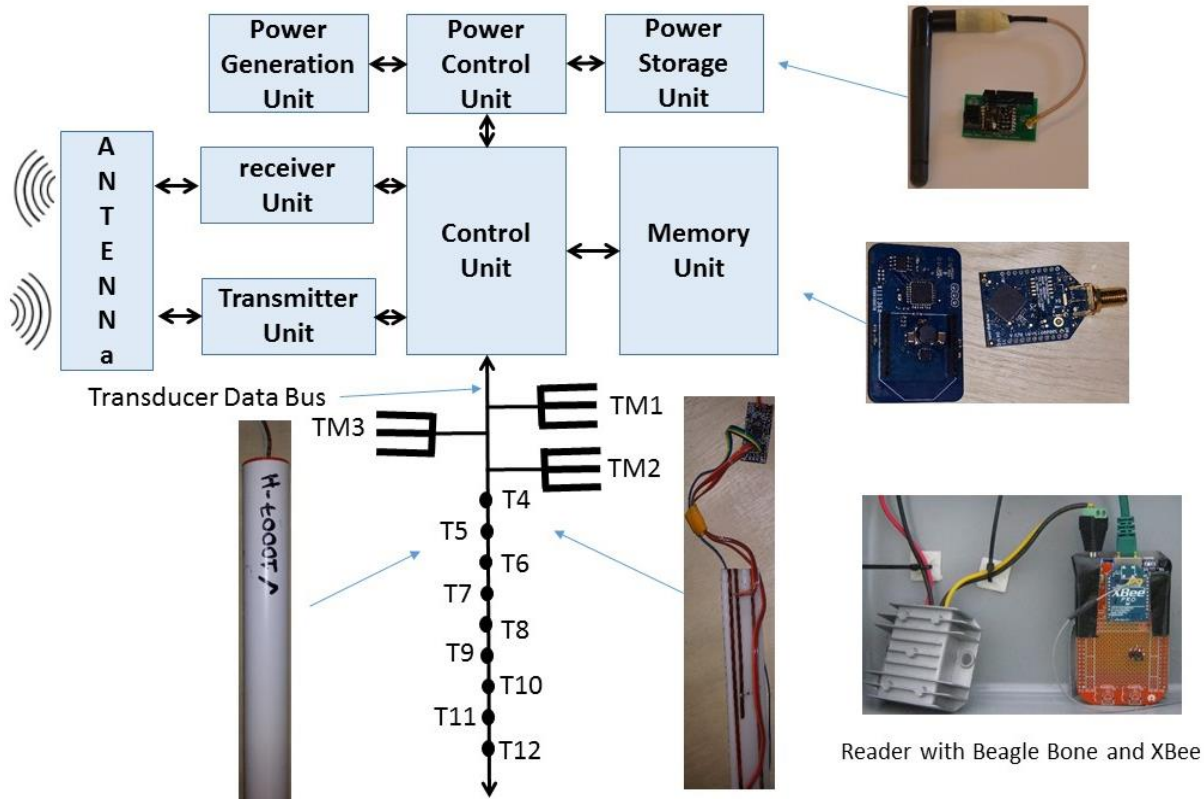


Figure 3: RF Sensor Node Architecture

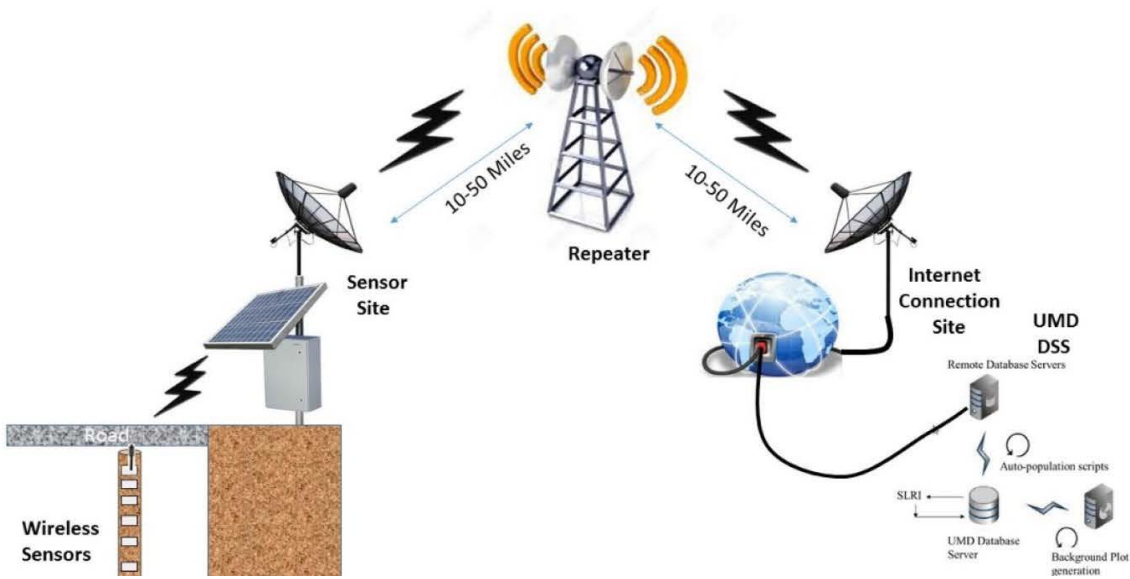


Figure 4: System data collection and communications for sensors and DSS

In the third sub task, 3b, members of the ECE and CEN team at UMD tested several fully integrated sensor(s), reader, logger, network elements and housings in a laboratory setting to validate operations in a wet soil environment.

## 2.2 Testing of Moisture Sensors

The Onset moisture sensor, model S-SM<sub>x</sub>-M005, was tested alongside the VH400. The tests used to compare the two sensors were conducted in a 6-inch diameter Proctor compaction mold. The tests included four iterations of moisture content, with two variations in compactive effort, for a total of eight combinations. The four iterations included dry soil, then soil with 3%, 6%, and 9% volumetric water content (VWC). The four moisture levels were tested in a relatively loose state, and then in a compacted state. The experimental setup, the testing procedure and results were described in detail in the MS Thesis of ECE student Zaidan F. Sheabar.

The results of the tests seem to indicate some inaccuracy for both the VH400 and the Onset sensors, although both sensors did exhibit relative trends to be expected. The VH400 did not achieve the accuracy of  $\pm 2\%$  listed on the manufacturer's website. Initially, it was suspected that there might be an error in the programmed piece-wise linear equations used from the VH400 website; however, this was verified to be false. The second culprit was suspected to be the strict use of integer math creating too many rounding errors, but the values fall well outside any normal range of error.

It is likely that the soil used in the laboratory affected the results to some extent. The type of soil surrounding the VH400 will affect its ability to read the moisture content of the soil, regardless of its sensitivity level. Most of these probes come “pre-calibrated” based upon manufacturers tests on a given soil type. Since these sensors are often used for agricultural purposes, the soils used by the manufacturer for calibration are often more of a silty/loamy soil appropriate for growing crops in. On the other hand, the soil used in the UMD laboratory compaction mold was classified as a poorly graded medium sand (USCS symbol SP). Most dielectric-based moisture content probes tend to work best when there is complete soil contact around the probe tines, typical of conditions in a fine-grained soil. When the probes are embedded in soils with a significant amount of medium to coarse sand and/or gravel sized particles, that contact is lost, and more scatter tends to be exhibited in the data measured by these probes. Since most highway base materials contain a significant amount of gravel, this continues to be a challenge faced when trying to monitor moisture regimes in highway applications.

### **2.3 Testing of Pressure Sensors**

An FSR pressure sensor was tested using a Durham Geo S-610 Load Frame in the UMD soils lab. A Plexiglas cylinder was filled ½ full with dry soil, at which point the pressure sensor was deposited face up, and then the remaining volume of the cylinder was filled with dry soil before being capped with a solid Plexiglas disk. The devices were powered on, and the load frame was set to increase load at a moderate speed. The smart node prototype was configured to take measurements once every 15 seconds, and a value was recorded from the load frame simultaneously. Measurements were recorded until 3 minutes had elapsed.

A graph of the load frame’s applied pressure versus output pressure of the prototype smart node FSR showed a large deviation between the pressure indicated by the FSR and the actual pressure applied by the load frame. The large deviation was attributed to:

- Lack of support underneath the FSR - the absence of a flat, smooth material behind the FSR can cause the output to skew as the active area may not fully actuate due to pebbles in the soil.
- Lack of a fixed location in the soil - as the force applied on the soil by the load frame increases, the soil begins to shift downward, also moving the FSR along with it, away from the optimal point for the testing.

It was therefore concluded that the FSR, as-is, will not be viable for accurate pressure measurements beneath roadways.

### **2.4 Calibration of Temperature Sensors**

During late summer 2015, a string of thermistors was calibrated in an apparatus developed by the ECE team to determine freezing temperature offsets. The sensors did not stabilize within a reasonable amount of time, and significant noise was observed in the data.

Several suggestions were made by the geotechnical group and by Frost Associates for modification of the ice bath apparatus and procedures used for the calibration. It is anticipated that implementation of these changes may result in better calibrations in the future.

The last sub task, focused on developing the UMass Dartmouth in situ RFID technologies roadway SLR test bed (Figure 5). The testbed system and sensor installations will undergo continuous testing for the foreseeable future to assist in debug, development of maintenance documents and in data for use in commercialization plans. The UMD testbed site and sensor data have been added to the UMD DSS.



**Figure 5: UMD Outdoor Environmental Testbed Facility**

## **Chapter 3 – Test site identification and deployment of sensor systems**

Identification of test sites for the project turned out to be a much more involved task than was originally envisioned. One of the main issues encountered was reluctance from the state DOTs to provide access to an internet router in a base facility to allow our wireless backend communications system to pass data through their system onto the web and ultimately to the UMD backend database system for extraction and use on the DSS system. After lengthy discussions and negotiations, one site was selected in Presque Isle Maine. To meet project goals of having multiple sites implemented, the Team also placed a full site on the UMD campus off of ring road in lot 17.

The sensor SLR test sites for Maine in Presque Isle (one sunny and one shady) were installed and recently upgraded to our most mature sensor and communications systems design. All systems are operating as designed and data collection and dissemination is ongoing. No additional sites were developed to leave time to improve the sensor systems physical designs and to lower construction costs to improve commercialization planning.

### **3.1 Conduct laboratory tests on soils from selected field sites**

To date, strings of wireless temperature sensors have been installed at two locations along Route 227 (State Road) in Mapleton, Maine. In November 2015, Maine DOT drilled two boreholes for those installations. Continuous soil sampling with standard penetration testing (SPT) was conducted, and the operations were logged by Maine DOT personnel. Boring No. HB-MAP-101 was located in a shady area adjacent to a low-lying evergreen swamp and a brook. Groundwater was encountered at about four feet below the top of pavement in that borehole. Boring No. HB-MAP-102 was located in open, sunny area on a small knoll near a crest in vertical alignment. No groundwater was encountered during drilling of that borehole, and auger refusal on probable bedrock was encountered at a depth of 13.4 feet.

A total of 18 soil samples were obtained from drilling operations (9 from the sunny site, and 9 from the shady site). Those samples were transported to the soils laboratory at UMD and subjected to the following standard index and classification tests:

- All of the samples collected were initially subjected to visual/manual identification procedures (dilatancy and toughness evaluation).
- The majority of the samples were then dried for moisture content determination and evaluation of dry strength (crushing) characteristics.
- Sieve analysis was conducted, and hydrometer tests and Atterberg Limits (liquid limit and plastic limit) were performed on the “fines” portion of selected samples, as appropriate.

Based upon the results of laboratory tests as well as visual/manual identification procedures, samples were classified according to the USCS and AASHTO systems.

A summary log for each borehole, which includes sample descriptions and classifications as well as results from Lab and field tests, was provided in combined “Deliverable #36 & #43” submitted on March 2, 2016. The grain size distribution curves (from sieve and hydrometer tests) are also included in Appendices A and B of that deliverable report.

## **Chapter 4 – Decision support system development**

The Decision Support System for Spring Load Restrictions implements a data aggregation web server to assist State Department of Transportation officials in their decision of when to temporarily restrict traffic on certain roadways during the spring thaw season. Atmospheric data is collected over publically accessible applications on the internet, such as WeatherUnderground.com, provided by third party weather stations across the states of New Hampshire and Maine. The atmospheric data is used to compute Cumulative Thawing Indexes (CTI) and Cumulative Freezing Indexes (CFI), both of which can be used in WWP and SLR protocols recommended by MnDOT (2014) to assist transportation agencies in deciding when to temporarily restrict traffic on certain roadways. In order to validate those recommended protocols, subsurface sensors installed by UMass Dartmouth report hourly subsurface temperature data at specific sites located in New Hampshire and Maine via wireless transmissions. CFI and CTI computations are compared with measured frost and thaw depths, respectively, to assess the recommended WWP and SLR protocols. In past seasons, the correlations have been strong; suggesting that use of CTI and CFI calculations (based off of atmospheric data) can be very helpful in deciding when to temporarily restrict traffic on certain roadways.

All collected data from sensors is imported into the web server. Data calculations are then run automatically, which determine the CTI and CFI values. The user can view all sites in the system, along with all collected data and calculated data values.

### **4.1 Limitations of the Old System**

During the spring of 2014, an analysis of the capabilities and library dependencies of the system was performed. It was found that while the project's current web site design was functional, it was limited in terms of functionality. When the application was first created, the technologies it used were current and relevant; however they have since gone out of date. Adding new features to the current system is burdensome due to the outdated technologies it utilizes. Previously, the system used CodeIgniter as the back-end PHP (server-side programming language) framework, 960.gs as the CSS (cascading style sheets) framework, and jQuery as a javascript library. While CodeIgniter works, there exist other tools which perform much better. New back-end PHP frameworks exist, and allow for faster development. 960.gs is a grid system initially released in 2008. While still functional, other CSS frameworks exist which offer much more powerful features and enable faster development. In essence, the web application's current implementation is built utilizing old technology.

Aside from the old technology, various aspects of the system required major refactoring. A large portion of the code base would need to have been rewritten in order to be more modular, and to accommodate the development of required features. Therefore, a new re-design from the ground-up using relevant technologies would yield an application which is much more efficient and faster to develop for, ultimately benefitting the project and its success.

## **4.2 History of the Transition from the Old System to the New System**

After much research, it was decided the new web application will be built with several frameworks and libraries. The back-end PHP framework is Symfony2, a high performance framework which is actively being developed and funded. For the front-end, the CSS framework is Bootstrap, a framework which allows responsive sites to be built that render almost universally across desktop computers, tablets, and phones. jQuery is also used. A third library used for the frontend is EmberJS, which is a Javascript MVC that allows scalable applications to be built. Many other libraries will be used, however these are the most prominent ones.

When taking into consideration features to be added in the future, the previous web site's design - both the front-end user interface and the back-end PHP framework - it was apparent a redesign was required in order to maintain efficient and timely development. The second version of the DSS ended up being a more powerful, scalable application.

## **4.3 Design of the New System**

### **A) Back-end Design**

#### **i) Architecture**

The system currently relies on Symfony2, a PHP framework, for the back-end. It retains a collection of entities (ex. a data collection site, a sensor reading), which are mapped to a MySQL database via an object relational mapper (ORM) powered by Doctrine. The ORM from Doctrine allows for the database to be abstracted to the point so that it can be modified automatically based on the entity a particular table represents. In addition to Symfony, several different third party bundles (a collection of classes, controllers, web resources, etc.) are utilized. Bundles are essentially plugins for Symfony which are made by third party developers, or the user. One could make their own bundle, and so far several bundles have been created to represent the sites, users, alerts, and the API (application programming interface). One third party bundle in use provides advanced serialization and deserialization of entities, allowing for interaction with the database via an API much easier and more advanced. Another bundle in use provides functionality to rapidly create an API for the system. The API allows the application itself (and third party applications, if chosen to) to interact with the server in a RESTful manner, ultimately yielding in specific data to be readily accessed, modified, and created through API calls. For example, an API call could return all collection sites stored in the system, with each site being a serialized representation of the entity existing in the MySQL database. The serialized representation of the site can contain information such as the site's name, location, etc. A third bundle currently installed generates documentation for the API, so all the endpoints available to call are presented in a user-friendly format. In short, the database back-end that retains the well-documented RESTful API accessible data provides not only an easier means to develop the application, but also could potentially pave the way for third party developers to build applications using the data.

Symfony2 has been configured to host the new DSS application. It is set up with the bundles described above, and is currently capable of hosting various bits of information. The application can create a user entity, with a name, email, and a state.



A site entity can also be created, with each site retaining a name, a state, latitude, longitude, elevation, the name of the data source, and an array of alert entities.

The application can also create an alert, which belongs to a site (giving it a reference to the parent site entity), containing a title, description, and a type. Currently, the back-end can validate requests to the API which either create or update a site. These validations check to ensure properties of an entity are valid (ex. ensuring the site name is not blank), and applying to any sort of interaction with the API.

In terms of the API, a given user can be retrieved by an ID. For sites, much more can be done. A list of all sites in the system can be retrieved, along with a specific site (given an ID). A site can also be created through the API, and a given site can also be deleted. Furthermore, a given site can be updated through the API. The current setup is powerful for if a property is added to the site, once the entity is updated in the Symfony application, the MySQL database will reflect the new property, and the API will automatically serialize the property along with the entity. Any validations that are added to the newly created property are also persisted and checked whenever the entity is validated.

Sensors were designed so they retain the sensor name (ex. atmospheric, pavement, etc.), the unit associated with the measurements it takes, and the average type (ie. whether to return a daily average or a daily cumulative of all readings). A sensor reading currently has a timestamp, a reading, the site it belongs to, and the sensor it belongs to. The site entity has also been designed to retain multiple sensor entities. By setting up the backend as so, a very modular flow has been established: a simple query can retrieve all sensor readings for a particular site, or even retrieve all sensors for a particular site. API endpoints have been established, which allow for simple GET requests to retrieve sensors or sensor readings for a particular site. A sensor reading average endpoint has also been created, which dynamically calculates either daily averages or daily cumulatives for all sensor readings, spanning all dates that the system has for that particular sensor type.

Functionality has also been added to associate sites with other sites. This allows for a particular site to be associated with another - when this occurs, the associated site will be passing along any sensor readings and sensors in API calls to the other site. The API calls are configured to contain a parameter which can be set to either return the associated data, or only the specified sites data. For example, if a WeatherUnderground site is associated with a DataGarrison site, a user that pings an endpoint to retrieve sensor readings and marks the parameter to retrieve associated data as true, the sensor readings in the DataGarrison site and the WeatherUnderground site will be returned. A feature such as this is quite powerful, for it allows for data from different sites to be joined together in a cohesive manner (ex. forecast data provided by WeatherUnderground can be joined with measured air temperature at one of our sites).

Data export has also been added in. If data for a particular site wants to be downloaded, visiting a URL will start the download of a CSV file containing every single sensor associated with the site, along with the corresponding readings between the specified dates. The same functionality has also been applied to data calculations - data calculations can also be downloaded as a CSV, containing the values for all calculations for a given site within the specified dates.

On the backend, additional data calculations were implemented in order to add CTI color indicators. These color indicators appear on the map, and signify the thawing state of each site. The data calculations calculate either the time till thaw, or the time since thaw, and then persists these values to the site entity within the database. These values are serialized alongside the site entity in the API, so the frontend can then incorporate these values in order to display the proper color. Another data calculation which was implemented was the date range for a particular site for which the site has collected data for. This is useful for it both speeds up data calculations due to caching the date range for which to perform calculations, but is also useful for displaying the available date range for the user.

## **ii) Email**

In terms of backend development, implementing email alerts remained another focus. The email engine chosen was Mandrill, which is a cloud based email system which enables the backend app to send emails via API calls. Mandrill enables for high volumes of email to be sent programmatically, making it the perfect tool for sending out DSS alerts to a subscription list. Mandrill also retains a very high uptime, and an almost nonexistent API error rate, making it a very reliable service to send emails with. Integration with the backend required the installation of a vendor framework, which abstracts the API calls to a more simple interface. The vendor framework required installation and configuration, along with an API token associated with a Mandrill account. Once the vendor framework was configured properly, creating the actual email templates began.

Creating email templates required integration of an external CSS framework in order to rapidly prototype a design. Due to the fact an external style sheet cannot be saved and referenced within an email, all email styles had to be converted inline, along with the framework. Therefore a tool was used to convert a source template, which used minimal markup per the framework, into an output HTML file containing all CSS styles inline. Once the design was in place, a means of dynamically adding in data to the content of the email had to be devised. Hence the Twig templating engine, which is natively utilized by Symfony, had to be leveraged in order to inject the dynamic content within the email. Thus the data could be passed into the email template, and Twig would automatically populate it with the appropriate data.

Once the template was created, it was then integrated into the sensor error detection command, which was created earlier. Once the command runs, and discovers sensor errors, it passes the information through Twig and a complete email template is created. From there, it is sent to the recipient(s). In terms of future development and capability, the email alerts can be expanded beyond just sensor errors - for it can include CTI & CFI alerts as well.

An internal email service was also created within the backend. This email service can be accessed from any controller or command within Symfony. Within the service, a function exists which can be invoked to send mail - which requires only a recipient email address, and the template along with the data which should also be used within the email. The email service further abstracts the means by which is required to send an email within the backend.

## **B) Production Application Deployments**

The entire application's directory structure has been revamped in order to maintain a more cohesive development process. The restructuring also improves the application deploy process to the production server - more on that later on. Involved with the restructuring was moving the Symfony2 framework and all DSS source code files, pertaining to the backend, to the root of the project directory. Then, the frontend Ember application was moved into its own bundle within the Symfony application.

In terms of the actual application deployment to the production server, much work was done in addition to the application's directory restructure. The production server had to be configured in a way so that it could host the DSS application, store all the data within a database, and also serve up the web application so that it can be browsed when visiting the website. Once the server configuration was all set up, a more robust deploy process had to be implemented.

In terms of the actual deploy process, an open source tool called Capifony was used. Capifony is a tool that allows you to run scripts on a server, and has a primary use of deploying Symfony applications. The deploy process was set up so that whenever a deploy is needed to be done, a single command is run. From there, Capifony SSH's into the production machine, pulls the latest code from the GitHub repository, and stores all the source code into a new directory. Next, Capifony runs all the necessary commands to configure the latest code to be run in the production environment. Finally a symbolic link is created from the path Apache is serving the website from, to the directory just created. Capifony also allows for the developer to rollback to a prior deploy if the latest deploy had something wrong with it. By implementing the deploy script, deploys are made much quicker, saving time.

## **C) Front-end Design**

### **i) Architecture**

The front-end is powered by EmberJS, which is a Javascript MVC that allows scalable applications to be built. Ember behaves more like a desktop application, so once the webpage loads, there is no waiting for pages to load - page transitions are instant meaning the only things that the user will have to wait to load is data asynchronously (ex. sensor readings). Ember is also powerful in that when paired with an API that presents serialized data such as the one currently implemented by the Symfony application, it is able to interface with the database with ease. Once the data is retrieved from the Symfony application via the API, the data is deserialized and persisted to models within the Ember application, which then can be rendered to the page in various ways. Handlebars, a templating library currently in use by this system, allows for properties of models to be rendered to the page dynamically. Changes can then be made to the model. Once the models are saved on the Ember application, Ember will automatically persist those changes to the server via the Symfony application API. In summation, both EmberJS and Symfony2 can be used in tandem to create a very powerful and scalable application.

All pages in the site have been styled with CSS, and uses the Bootstrap CSS framework. So far the entire site is responsive due to using Bootstrap, meaning that it renders similarly on desktops, laptops, tablets, and phones. This allows for the same functionality to be available on the site on all screen sizes. In its current state, the site can be navigated and used like any other webpage.

The Ember application has models for users, sites, and alerts at the moment, which allow the serialized API data to be stored within the application. Both user and site information are loaded from the back-end and are stored in these models. It interfaces with the Symfony application to pull in database data. All the sites are displayed on the map screen in a similar way the current system presents the map. Users can transition to a specific site from the map screen, which brings up the site view.

The frontend application also accommodates the display of sensors and sensor readings, with both sensors and sensor readings being sideloaded whenever the data is needed. Currently, when the user visits the ‘Collected Data’ page of a site, all the sensor and sensor reading data is fetched and loaded into the frontend application, where it is displayed in either a graph (Figure 6) and / or table. The system is smart enough to know that if you leave the “Collected Data” page of a particular site, navigate to another page, then come back to that same page, it will not fetch the data from the server but merely pull up the data previously retrieved. The graph itself is very modular: it allows for the user to zoom in on specific date ranges, and the user can even change the type of graph on the fly (ex. line, bar, scatter, etc.). When the user hovers over a certain data point, the date and reading are presented in a box, giving detail about the currently hovered data point. Currently the user views one type of sensor at a time on the ‘Collected Data’ page. When the user goes to switch the sensor type, both the graph and table have their values dynamically updated to reflect the new data for that specific sensor. The sensors available are automatically loaded in based on what the sensor entities are associated with the site.

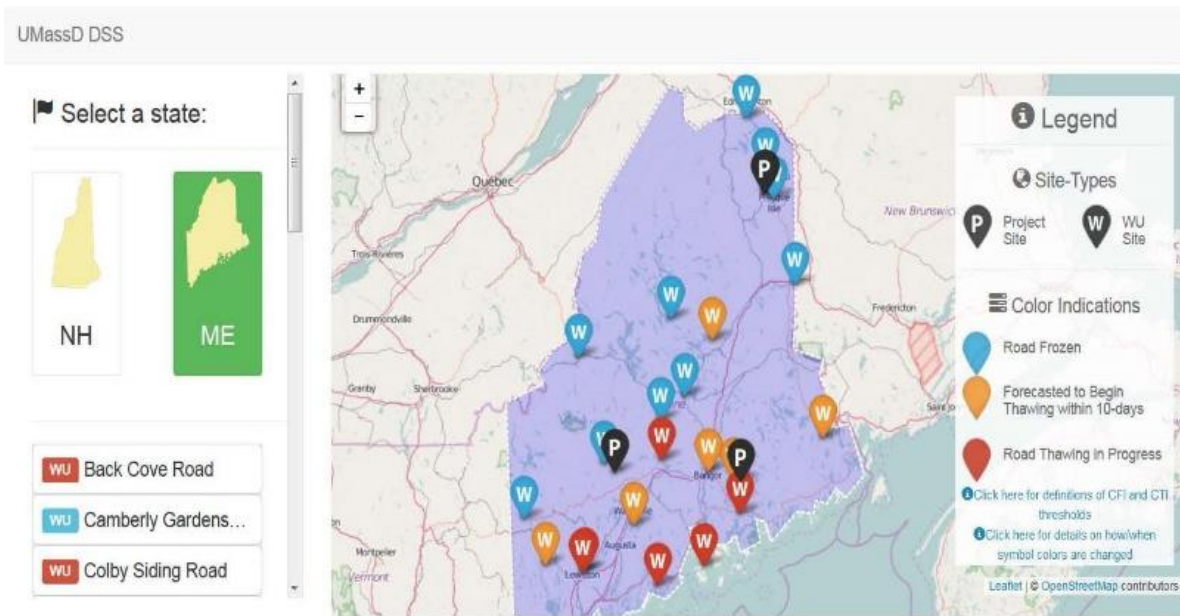


Figure 6: Graphical DSS rendering of spring thaw progression

The predictor tools page was also implemented with a graph displaying CTI calculations, date range picker, and also a refresh button. On the graph itself, it is formatted so that the thawing event can be viewed more easily. There are two regions on the graph - red and green. The region below the threshold is green, whereas the region above the threshold is red. Such a format allows the user to very quickly discern whether or not an event is occurring.

Changes were also made to the sensor selector component. Whenever a sensor group did not have any sensors active for a particular site, the sensor group is now hidden for that site until data becomes available for it. Additionally, whenever a sensor group had a collection of sensors that each retain a depth, they are now ordered (in ascending order) according to their depth. The sensor group which pertained to all the air temperature readings has been reworked to read either 'air temp (°F)' or 'subsurface temp (°F)' in order to be more clear on which data the sensors are reporting. Additionally unit depth markings were put on the end of sensors to indicate the specific unit of depth for the data (e.g. subsurface-36" as opposed to subsurface-36). Lastly, whenever a site does not have subsurface data (e.g. a WeatherUnderground site), the subsurface sensors tab will be hidden.

## **ii) Dependency Library Updates**

The frontend Javascript MVC framework, Ember.js, had to be upgraded to the latest version several times during the development of the project, for new versions would continuously be release. Not only did the updates result in performance improvements in the application, but more recent versions of Ember.js have included a new command line utility and build process called Ember-CLI. The most significant gain from the new CLI pertains to allowing the Javascript code to be written in a true module approach. Such an approach is called ES6 Module Syntax. The CLI will then compile this syntax to the plain Javascript, which the browser runs. This is an improvement from before compared to using Grunt to build the frontend application, which was much slower building the application. With upgrading the frontend we also secure a more recent version of the framework, along with better backward compatibility with future versions of Ember.

When the frontend was upgraded, it required large portions of the code to be revamped and optimized due to new design patterns and deprecations within the frontend framework. Aside from the stated benefits of the latest version of the frontend application, all pieces of code were examined and refactored - leaving it more optimized and readable.

Another motivation for refactoring the code was due to how slow the graphing component could be. A large portion of this pertains to the fact that the old graphing software utilized SVG to render the graphics. While it worked and looked nice, when many data points were loaded into the graph, it was slow to render and navigate. At times, the page would appear to freeze when loading in a large date range of data, which leads to a negative user experience. New means are implemented in order to avoid slowdowns and page hanging, primarily with using alternative rendering technology. A new open source graphing library is now used which is based on HTML5 canvas, offering better performance.

Consequently, by using this new rendering technology, we will be able to render more data points faster. If necessary, a portion of the data sorting done by the frontend application when graphing can be offloaded onto the backend server in order to improve performance.

In addition to performance increase, the user interface has been examined and is in the process of being redesigned. Many aspects of the frontend application - from the state selector to the sensor selector - can benefit greatly from redesigns in order to make them easier to use. By examining difficulties with these particular aspects of the site, new designs have been drafted and are beginning to be implemented site wide. Additionally, performing the redesign during the offseason is more desirable, as more focus can be devoted to it.

#### **D) Data Import**

Extensive work has been completed in an effort to create a modular method of importing data supplied by Data Garrison. A compact algorithm has been composed that automatically retrieves mapped data values for any Data Garrison site in the system. The algorithm checks every single sensor reading for every single date, and makes sure that the system is up to date - while being smart enough not to add duplicate readings.

WeatherUnderground data has been implemented, which includes fetching the previous' days average air temperature, and the 10-day forecast air temperature. The current implementation allows for a specific command to run daily, which pulls in both the measured and forecast data for all WeatherUnderground sites in the system.

Parameters have also been added that allow for specific date ranges to be imported from various sources, along with a data check to verify if the data had previously been imported. For one of our data providers, Warren Flats, an auto-mapping feature has been established which automatically links sensors in the database to the column names in the data, which allows for dynamic data importation. Such a feature safeguards against the order of the data columns changing, for the system will automatically adjust.

The sensor reading average algorithm has also been improved (for computing the averages of imported data), speeding it up around 60%. The system can now process much greater volume of sensor readings, at a faster rate. An algorithm has been implemented for calculating CTI and DFI.

Another feature that has been implemented pertains to the ability to make adjustments to sensor reading values, like the water height sensors for Mariaville and Warren Flats. The adjustments for the sensors at these sites are made according to the depth to groundwater table equations. This feature designed such that the original data is preserved in the system - the computations are made and the calculated values are stored so that they can be viewed in the web application.

A method to detect imported sensor reading errors has been designed and implemented on the DSS. Such a system will scan through the most recent data measurements taken at the project sites and if any of the data points are deemed as inaccurate or suspicious, which would signify a faulty sensor, an alert is created and can be viewed via the web application.

For now the system utilizes basic sensor error detection (ex. a subsurface sensor reports duplicate values for more than 24 hours). In the future, the means by which wthe Teamdeem a set of sensor readings as inaccurate can be easily expanded upon. This sensor error detection system can also be expanded to send emails to the project members, alerting them of the faulty sensor.

Also in terms of data import, an additional feature has been implemented which pertains to retrieving historical data information from Weather Underground. The feature is implemented as a command in the backend system, so it can be run by the system administrator or on a schedule. When initiating the command, the site, along with the start and end date for the data to retrieve are required in order to begin the data retrieval. Once these parameters are specified and the command is run, the system automatically retrieves all Weather Underground data for the sites within the dates, which makes backlogging data very easy for any particular site within the DSS. In this way a site can be added at any point in time, even if it is in the middle of the frost / thaw season, meaning the system can still retain a comprehensive collected data set at all times. Furthermore, if data for a day or even multiple days is not available for due to a delay with Weather Underground, the system can attempt to retrieve these missing data points at a later point. Hence, the capabilities of the DSS are expanded.

#### **4.4 New System: Additional Features**

##### **A) Data Calculations**

The top and bottom frozen data calculations has also been implemented. Like on the previous version of the DSS, a graph was located on the predictor tools page which the users could view both the depth and how much was frozen for subsurface sensors. The new system is now able to calculate these data points based on the subsurface temperature data provided by the current active project sites. In the future, these data points will be presented on the ‘Predictor Tools’ page on a separate graph for every project site, and will help verify the validity of the CTI calculations being performed.

Additional data calculations for the Model 158 have been implemented. These calculations are site specific, and account for the different thermal properties of up to 5 soil layers. So far, these calculations are provided for both the Warren Flats and Madison sites, and run daily alongside the other data calculations.

## Chapter 5 – Evaluation of SLR forecast models and systems

The UMD team with support from Frost Associates continued with evaluation of various models and protocols for use in SLR timing. In the previous RITA study, the team evaluated and endorsed a CTI threshold protocol (originally developed by MnDOT) for SLR application. One of the advantages of this protocol is that the only input is air temperature, and it does not require any specialized site-specific calibration. The NH and Maine state DOTs strongly support adoption of this protocol, and the UMD DSS has been updated to include a graphical presentation of that CTI computation and threshold (at the 3 instrumented test sites, as well as at several additional locations supported by weather-underground).

While several CTI thresholds for removing SLRs have been suggested in the literature, the UMD team concluded that there was not currently enough evidence to support any of those thresholds for SLR removal without further evaluation and/or calibration. Indeed, the decision to remove SLRs is much more complex, and is hampered by the wide variation that exists in subsurface soil and moisture conditions. Therefore, to assist in SLR removal decisions, more complex models capable of accurately predicting frost and thaw depths are desirable. During the course of this contract, the UMD team investigated the following models for predicting frost and thaw depths from the onset of freezing through the completion of thawing:

- A. a modified version of the US Army Corps Model 158
- B. a freeze-thaw index model, requiring site-specific calibration
- C. the EICM model (embedded in the AASHTO MEPDG software)
- D. the FASST program (developed by CRREL)
- E. the TEMP/W program (included in the GeoStudio software suite)

A summary of each of the evaluated models is included below, followed by recommendations:

- A. Processing of the 2013-2014 data in the Modified Model 158 was completed for 3 sites that were instrumented as part of the previous RITA project. Plots of model output were included in Appendix A of Deliverable 13. Data from 2014-2015 was also processed for those 3 sites, but instrumentation from 2 of the 3 sites went down in 2015, so 2015-2016 data was processed only for the Warren Flats, NH test site. Those plots generally showed very good agreement between measured frost and thaw depths and those predicted by the model.
- B. Calibration of a site-specific freeze-thaw index model and processing of the 2013-2014 data in that model was completed for the 3 instrumented sites. Plots of model output were included in Appendix A of Deliverable 13. Data from 2014-2015 was also processed for those 3 sites, but (as noted above) instrumentation from 2 of the 3 sites went down in 2015, so the freeze-thaw index model was only evaluated at the Warren Flats, NH test site for 2015-2016.



Those plots generally showed very good agreement between measured frost and thaw depths and those predicted by the model.

- C. Preliminary evaluation of the version of the EICM embedded in the Pavement ME Design software (MEPDG) was conducted by Christopher Cabral (Frost Associates). The 3 evaluations initially conducted using the Pavement ME Design software were for Lake Tarleton (LT) and Warren Flats (WF) during 2009-2010, and Mariaville during 2013-2014. Plots of model output were included in Appendix C of Deliverable 13. The hourly climatic data for LT and WF was obtained from the Davis weather stations used at both of these sites during previous NH DOT/FS studies. Any small gaps in the data (i.e. a few hours with blank values) were filled with data by linearly interpolating between the values before and after the gap. The Mariaville 2013-2014 climatic data was obtained from instrumentation installed from the previous UMD/RITA project, and had a significant amount of missing data. Working notes describing the data reduction used to fill in the missing gaps are included in Appendix C of Deliverable 13.

Subsequent to the initial work conducted by Christopher Cabral, Richard Berg (Frost Associates) performed numerous additional simulations for the LT site using this software, but varying some of the input parameters. Results from Berg's simulations were not substantially different from Cabral's, and were described in detail in a report entitled, "Supplemental Deliverable for Task 7: Evaluation of SLR Timing Forecasting Models," submitted October 26, 2014.

Based upon the review of the EICM embedded in the MEPDG, it was concluded that the MEPDG based software was not an appropriate tool for use in SLR application and/or removal due to its inability to accurately track the thawing process, as well as the fact that it was not a user-friendly tool for frost-thaw depth modeling.

- D. Advisory board member Sally Schoop provided the UMD research team with a copy of the CRREL FASST software. The research team attempted to run this Fortran code on a windows based computer, but found that it required significant programming efforts by someone with computer science expertise to do so. The input files must be constructed according to a specific Fortran format, which is very unfamiliar to most Civil Engineers, and so this also required assistance from a computer science graduate student.

In any case, two sets of simulations were run using that software for the Lake Tarleton site in NH during the 2009-2010 freeze-thaw season. A detailed description of those simulations is provided in a report entitled, "Supplemental Deliverable for Task 7: Evaluation of SLR Timing Forecasting Models," submitted October 26, 2014. Although the FASST model is theoretically very robust, the predictions obtained from the two simulations did not compare well with measured subsurface temperature data. Additionally, for most Civil Engineers (with little to no Fortran background), the model is not very user-friendly to run. As such, the research team concluded that additional work with the FASST model was not warranted for this project.

- E. The research team also reviewed work recently conducted at Lakehead University (Ontario, Canada) using the finite element program TEMP/W.

- F. The latest version of that software enables prediction of frost and thaw depths using 2 alternative approaches. One is a sophisticated, data intensive approach, where a multitude of input parameters are required (air temperature, solar radiation, wind speed, relative humidity, etc.). The other is a more simplified approach, using measured or predicted air temperatures as the only input, and applying n-factors to transfer air temperatures to pavement surface temperatures (for the upper boundary condition). Because the producer of this software was keenly interested in the research being conducted under this contract, they provided a free 1-year license for this TEMP/W software to the research team. The research team utilized a comprehensive set of data collected previously at Lake Tarleton, NH as input for this software, and evaluated the TEMP/W model during year 2 of this contract.

Results of that evaluation were described in detail in a report entitled, “Supplemental Deliverable for Task 7: Evaluation of TempW Model” submitted October 26, 2014. It was found that, while the Temp/W model predicted frost-thaw profiles fairly well for two seasons (2008-2009 and 2009-2010), the predicted profiles did not match measured profiles as well for the other three frost-thaw seasons. In 2010-2011 and 2011-2012, the model reasonably tracked the final thawing profile but did not track freezing very well. And in 2007-2008, the model predictions did do a good job in matching either measured frost or thaw depths. The overall conclusion was that the Temp/W model is not the best choice for predicting frost and thaw profiles beneath roadways for use in SLR timing. The conclusion was based, in part, on the discrepancy between measured and predicted frost and thaw depths, and in part due to the proprietary software that is required. Many of the DOT personnel charged with making SLR posting decisions are not familiar with that software, and would not likely want to invest the time and resources that would be necessary to utilize that software for SLR timing.

In summary, in terms of numerical models for predicting frost and thaw penetration, the research team concluded that the two models described in items C and D above are not appropriate candidates for use in SLR application and/or removal due to their inability to accurately track the freezing and/or thawing processes, as well as the fact that they are not very user-friendly. The numerical model described in item E performed somewhat better in terms of tracking the freezing and/or thawing processes, but was still not given highest priority for utilization.

In terms of other models for predicting frost and thaw penetration, both the Modified Model 158 and the site-specific freeze-thaw index model showed much promise. In general, results from both the Modified Model 158 and the Freeze-Thaw Index Model showed very good agreement between measured frost and thaw depths and those predicted by the model. The main deviation for both models was that they predicted frost depths slightly deeper than measured depths, and they predicted end-of-thaw dates slightly later than measured. Ultimately, the Modified Model 158 was selected for incorporation onto the UMass DSS. Although this model does require information from a soil boring at the site to determine material types and layer thicknesses, it generally does not require additional site-specific calibration(s).

The Freeze-Thaw Index Model, on the other hand, requires one or more years of measured frost and thaw depths at a site in order to determine the required coefficients for that model. While continued development of the Freeze-Thaw Index Model may lead to more “generalized” coefficients (and thus eliminate the need for site-specific calibrations), it was felt that the Modified Model 158 was currently the better candidate for incorporation onto the UMass DSS.

## **Chapter 6 – Commercialization plan development**

The UMD team and commercialization partners continue discussions on product developments and commercialization efforts. The UMD Team's draft marketing and business plans was delivered under report number OASRTRS-14-H-UMDA Deliverable #65.

With support from the UMD Team's partners (INEX, TRUERoads, DBN IoT, TranSense) the team continues searching for commercialization funding beyond present funding under the OASRTRS-14-H-UMDA agreement. An agreement with US Forest Service resulted in 3 new states and 5 new testing sites being added into the system. It is anticipated that through this development additional sources of commercialization interest and funding will be uncovered.

The UMD Team continue working with commercialization partners to disseminate the commercial benefits of the DSS SLR system to support transportation management problems. The UMS Team has engaged with numerous special interest groups working on commercialization, and technology enhancements supporting efforts tangential to our specific project goals with the hope of furthering the commercialization of the projects products. The UMD Team continues to investigate additional funding to support the road towards commercialization.

### **6.1 Project Dissemination, Outreach and Commercialization**

The projects web portal is operational. Content for the site is being added to enhance commercialization efforts. The Project site - [slr.umassd.edu](http://slr.umassd.edu) will continue be enhanced as products are developed and marketing materials become available.

The UMD team continues to meet with present and potential commercialization partners (TruRoads LLC, DBN IoT, Global Insights, TranSense, etc.), and for infrastructure communications support (SeNet, Ubiquiti, Tycon, etc.). The UMD team continues interacting with these organizations to facilitate commercialization efforts.

Dr. Paul Fortier working with US Forest Service added 5 additional test sites supporting US Forest Service unpaved and paved road management. The new test sites are in New Hampshire (2 sites), Pennsylvania, and Oregon State. The Idaho DOT has shown interest in incorporating our system into their ongoing smart roadway projects. The additional roadway sensor sites and exposure resulting from these sites will enhance the projects outreach and more importantly commercialization goals.

The UMD Team continues to work with state DOT's and US Forestry to refine SLR setting and removal criteria. We continue working with commercial groups and organizations to define what the market is for underground sensors, the UMD DSS and forecasting tools and how these can be developed into commercial products.

This includes working with partners to develop return on investment stories for potential clients and customers.

## Chapter 7 – Conclusions

The project as a whole was quite challenging due to the involvement of multiple engineering disciplines and numerous governmental, state and local agencies. Not to mention private companies that have become involved as the UMD Team moved towards commercialization. The initial challenges associated with working with multiple engineering disciplines involved getting everyone to agree on notation and language. Each discipline has their own acronyms and even different meanings for numerous entities and concepts used within the project. The TAC meetings and regular team meeting alleviated many of these misconceptions and problems early.

Similar challenges were experienced in getting buy-in from state DOT's and partners, since they too had different understanding of the various technologies and concepts that were being researched to develop and insert into their working environment. Coordination among the various potential partners was challenging.

The development of the sensors from initial experimentations and prototypes proved to be much more involved, time consuming and costly than originally planned for. Sensor development went through 4 iterations, with each new design taking lessons learned from the prior iteration and improving upon them. The final design proved to work above expectations, with read ranges of 125 feet away from the embedded sensor proving to be quite straight forward. The UMD Team worked hard to resolve issues with the cost and distance issue between the street level data extraction point and the data sink point within a DOT facility. In the initial designs, continuously connected RF dish systems were used. The dish communications system proved to be robust, but the power requirements were excessive. The solar power systems cost became the overwhelming cost. To alleviate this cost, the Team moved towards a communications system that could be connected as needed, using recently released COTS radio frequency communications nodes. These RF communications nodes proved to be orders of magnitude cheaper and performed nearly at the same level. The final system design uses these enhanced RF communications nodes and are being worked into cost models.

The DSS system improvements have made the user's system more robust than the first iteration delivered during phase one of this project. Within the new DSS system, making changes to sites or visual appearance are fairly easy and extending facilities are quite simple using modular design insertion concepts. The system now extracts information from multiple external sources as well as from the project database to render a display that quickly conveys to the user the state of roadways within a geographic region. The site also supports the drilling down to finer levels to allow users to focus on their specific region or roadways, enhancing the ability to place and more importantly remove SLRs. The DSS system also provides the user the capability to simply extract and download data sets for a specific site and sensor configuration allowing for ad hoc and user specific analysis of data.

None of these capabilities would be useful without the SLR forecasting models that have been integrated into the DSS system. The numerous forecasting tools were ported onto the site to enhance the support of SLR planning activities.

The team examined a variety of existing frost-thaw forecasting models, while at the same time examined new and evolving models to determine which models were best suited to the problem of SLR placement and removal. The modified model 158 has proven to be one of the best in forecasting the SLR activities of interest to our users. Our team decided to take the approach used by the weather forecasting community. The UMD Team selected many models and use them all to produce multiple solutions that users can utilize in determining the best scenarios to place or remove SLR. If additional funding can be found, the Team would like to combine the various models and develop a composite “best solution” that like weather forecasting can be used to make a best guess forecast for use by customers.

The final component of the project dealt with commercialization. Commercialization was a requirement of the project from the beginning. Commercialization along with design and development of products proved to be much harder than originally planned. Commercialization required the development of possible funding sources to seed product refinement, advertising, and roll out along with company funding to support engineering, marketing, sales as well as customer service positions. The development of the commercialization marketing and business plan required almost a continuous involvement with outside entities and the UMD team’s business school. Engagement with various funding sources such as angel funding, and other sensor and roadway management companies took on an increased importance during the team’s extension year. The UMD Team has nurtured engagement with 4 primary companies that may support further commercialization of the project’s products. The outcome of these engagements will depend upon future joint funding source development as well as from constructive feedback from users and early adopters of the UMD Team’s technology.

## References

- [1] Bradley, A.H., Ahammed, M.A., Hilderman, S., and Kass, S. (2012). *Responding to Climate Change with Rational Approaches for Managing Seasonal Weight Programs in Manitoba*. Proceedings of the American Society of Civil Engineers 15<sup>th</sup> International Conference on Cold Regions Engineering. CD-ROM. Quebec City, Canada, August 19-22, 2012.
- [2] Minnesota Department of Transportation (Mn DOT), “*Policy and Process for Seasonal Load Limit Starting and Ending Dates*,” Minnesota Department of Transportation, Policy, Safety & Strategic Initiatives Division, Technical Memorandum No. 09-09-MAT-02, June 29, 2009.
- [3] Embacher, R., "Duration of Spring Thaw Recovery for Aggregate-Surfaced Roads," Transportation Research Record: Journal of the Transportation Research Board No 1967, pp. 27-35, Transportation Research Board of the National Academies, Washington D.C., 2006.
- [4] Hanek, G., M. Truebe, and M.A. Kestler, Using Time Domain Reflectometry (TDR) and Radio Frequency (RF) Devices to Monitor Seasonal Moisture Variation in Forest Road Subgrade and Base Materials, U.S. Department of Agriculture, Forest Service, San Dimas Technology and Development Center, San Dimas, CA, 2001.
- [5] Eaton, R.A., A. Hanscom, M.A. Kestler, A. Hall, H.J. Miller, and R.L. Berg, "Spring Thaw Predictor and Development of Real Time Spring Load Restrictions," Proceedings of the American Society of Civil Engineers 14th International Conference on Cold Regions Engineering, Duluth, MN, August 30-September 2, 2009<sup>b</sup>.
- [6] Eaton, R.A., R.L. Berg, A. Hall, H.J. Miller, and M.A. Kestler, "Initial Analysis of the New Hampshire Spring Load Restriction Procedure," Proceedings of the American Society of Civil Engineers 14th International Conference on Cold Regions Engineering, Duluth, MN, August 30-September 2, 2009<sup>c</sup>.
- [7] Kestler, M.A., Richard L. Berg, Bryan C. Steinert, Gordon Hanek, Mark Truebe, and Dana N. Humphrey, *Determining When to Place and Remove Spring Load Restrictions on Low Volume Roads: Three Low-Cost Techniques*. Low Volume Roads Conference, Austin, TX, Transportation Research Record 1989, pp 219-229, WA, DC. June 2007.
- [8] Ashton, Kevin: *That 'Internet of Things' Thing*. In: RFID Journal, 22 July 2009. Retrieved 17 December 2012.
- [9] Dziadak, K., Sommerville, J. and Kumar, B. *RFID based 3D buried assets location system*. J. Information Technol. Construction, 2008, **13**, 155–165.
- [10] Faridazar, F. and Lajnef, N. *Intelligent multi-sensor measurements to enhance pavement monitoring and safety*. In Passive Wireless Sensor Tag Workshop. NASA, Houston, 2011.
- [11] Roberti, M. *RF sensors could optimize crop irrigation*. RFID J. Mag. Online. <http://www.rfidjournal.com/magazine/article/8172>, accessed 12.17.2012.
- [12] Marquis, B., *Mechanistic Approach to Determine Spring Load Restrictions in Maine*, Technical Report No. 08-1, Maine Department of Transportation, Bangor, Maine, 2008.
- [13] Ovik, J.M., J.A. Siekmeier, and D.A. Van Deusen, *Improved Spring Load Restriction Guidelines Using Mechanistic Analysis*, Technical Report, Minnesota Department of Transportation, 2000.



- [14] Eaton, R.A., M.A. Kestler, and A. Hall, *Spring Thaw Predictor & Development of Real Time Spring Load Restrictions, First Two-Year Data Report*, SP&R Research Project No. 14282K, New Hampshire Department of Transportation, Concord, NH, 2009<sup>a</sup>.
- [15] Franek, O., Kyritsi, P. and Pedersen, G. F. *RFID on the road – some considerations about passive tag antennas*. In Proc. 5th European Conference on Antennas and Propagation (EUCAP). Rome, 2011, 1203–1207.

