A Model Based Systems Engineering Approach for Behavioral Responses to Advanced Quantitative Precipitation Information

William Brooks
Systems Engineering Department
Colorado State University
Ft. Collins, CO
brookswm@colostate.edu

V. Chandrasekar
Electrical and Computer Engineering Department
Colorado State University
Ft. Collins, CO
chandra@colostate.edu

Greg Pratt
NOAA Global Systems Laboratory
Boulder, CO
greg.pratt@noaa.gov

Rob Cifelli
NOAA Physical Sciences Laboratory
Boulder, CO
rob.cifelli@noaa.gov

***Abstract*— Atmospheric rivers (AR) contribute a significant portion of the precipitation for the United States' west coast. Flooding events caused by ARs can create millions of dollars in damages. Identifying probable ARs and early warning of potential flooding events can provide state and local agencies the time to prepare for such events. The Bay Area Advanced Quantitative Precipitation Information (AQPI) system links users with varied meteorological data ranging from weather radar observations, short term forecasts (Nowcast), 12-hour forecasts, and coastal inundation modeling data. The range of data types presents an opportunity for proactive responses by the users; however, it also presents the challenge of ensuring the correct data is available to the appropriate user. Through Model Based Systems Engineering (MBSE) Behavioral Analysis, the AQPI team analyzes the appropriate requirements for each of the different types of AQPI users. MBSE behavioral analysis leads to the development of a system that services the broad user community's needs while giving each user group a specific interface tailored for them. This engineering approach also allows for the separation of the processing and weather model execution from the user interface. This separation allows for development and advancements in processing without being tied to the user interfaces.**

Keywords—MBSE, AQPI, Systems Engineering

I. Introduction

To develop a radar network evaluation system, reliable data covering a broad range of sources is necessary. Data are required to enable decision-makers the ability to make informed decisions. The current approach involves web scrapping and individual data mining per organization. Having a system that provides data and modeling capabilities for both forecasts and longer-term products enables users to have a single source for information, reducing time to make decisions.

II. Background and Theory

The Bay Area Advanced Quantitative Precipitation Information (AQPI) system covers the area around San Francisco, Contra County, and Sonoma County. In addition to the National Weather Service radars, satellite imagery and locally controlled X-band radars are the active sensors of the Bay Area AQPI. Each county provides a set of stream gauges, moisture sensors, and rain gauges. The Bay Area AQPI utilizes this information as well as forecast models and coastal flooding model COSMOS [1].

Model Based Systems Engineering (MBSE) utilizes a model to maintain the different documents and diagrams necessary to describe a system, requirements, and expected behaviors. Starting with the various users, the system engineer begins to define the systems' expected behavior [2]. To communicate between groups and users, the systems engineer utilizes the Systems Modeling Language (SysML).

MBSE provides an approach to address multiple users' needs and requirements for a system while understanding the impacts of changes. AQPI, with its numerous users with various needs, is an excellent opportunity to apply MBSE to the development and subsequent deployment of AQPI.

III. Literature Survey

The west coast (California, Oregon, and Washington) receive 30-50% of all precipitation from a phenomenon commonly known as an Atmospheric River (AR), narrow belts of concentrated moisture with an average width of 400-600 km [3]. With such intense precipitation events, ARs often cause flooding. From October 1997 through January 2006, eight damaging flooding events on the Russian River are attributed to ARs [4]. Detecting and predicting ARs is vital to water management, as well as municipal and emergency managers.

Observations and raw data only provide a snapshot in space and time. Filling in the gaps and making a continuous picture of the weather comes from the Multi-Radar Multi-Sensor (MRMS) system. Combining radar data from the conterminous United States (CONUS) and southern Canada with satellite data, meteorological sensor data, and atmospheric environmental data, MRMS provides a quantitative precipitation estimation (QPE) from 20° to 55°N Latitude spanning 60° to 130°W Longitude [5].

AQPI systems provide a user community with an interface to create tailored products from a consistent set of data derived from various sources, including weather satellites, radars, sensors, and models [6]. During the development of the AQPI system, the data sources are identified, including new sensors and radars specific to the AQPI, as well as the nationally available information. During operations, users define the products required, timeliness of the products, and alert conditions. The AQPI maintains the data sets based on different factors including processing needs, update frequency, and data variability.

IV. Contribution: Behavioral Analysis Gap

The AQPI system offers the potential estimated average savings of $61million a year through flood damage mitigation, water supply, ecosystem, recreation, and transportation [7]. Unfortunately, the benefits can only be realized if the information is in the hands of the people making the decisions. If the behavior of the AQPI does not align with user expectations, then no one will use the tool. Behavioral analysis includes not only mathematical and data responses but also the user interface experience. This paper uses the systems engineering approach to investigate the different expected behaviors of the AQPI.

V. Approach

Using Cameo Enterprise Architecture (EA) to model the system, including user interfaces, the AQPI team identifies the desired behavior for the system. Cameo EA builds a model of the system and allows for calls to external simulations or canned data depending on the type of analysis being performed.

Not only does the simulation display the user interfaces, but it also captures the underlying exchanges and tests controls and constraints on the data to ensure realistic behavior. Full model behavioral analysis ensures responses are representative of the operational AQPI.

A. Metamodel

The Bay Area AQPI domain covers several aspects including observation retrieval, data processing, forecast models, and user interactions. A metamodel provides a blueprint of the components and relationships, and it serves as the foundation for developing the model. Fig. 1 presents the metamodel for the domain.

Counties contain sensors that provide observations. Observations help drive weather models in developing forecasts, which also force model development of subsequent and longer forecasts. Users interact with the AQPI domain to make informed decisions based upon forecasts and observations.

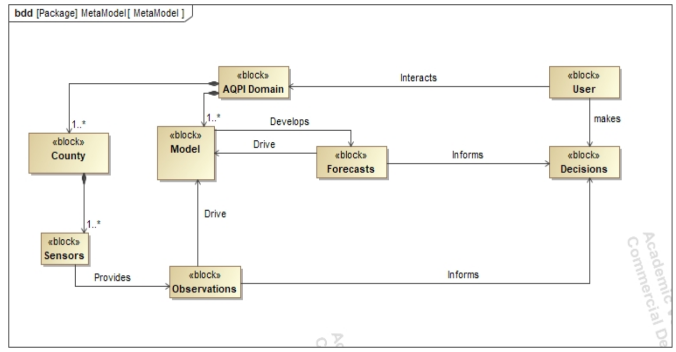

Fig. 1 AQPI Metamodel

B. Model Structure

Based upon the Meta Model, the creation of the Bay Area AQPI executable model follows.

1) AQPI Context

A context model defines the parts of the system, including the actors (external influencers), using a Block Definition Diagram (BDD). A block is a SysML concept which represents a structural unit with distinguishing features [2]. Blocks represent a myriad of items from software and hardware to the environment. All parts of the Context are represented by a block, with one of the blocks being the system of interest (Fig. 2). Each component can be further decomposed as part of the model or in a separate effort. This allows for focusing on development and keeping distractions down. Additionally, the system engineer can treat components as black boxes except for the one which is being analyzed.

2) Data Exchanges and Interfaces

In SysML, data exchanges are captured using Ports. A port enables the systems engineer to define what information moves

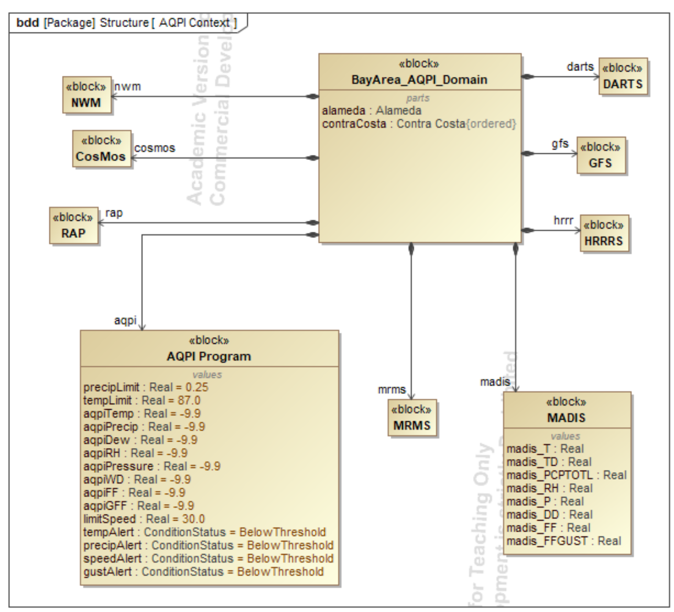

Fig. 2 APQI Context Diagram

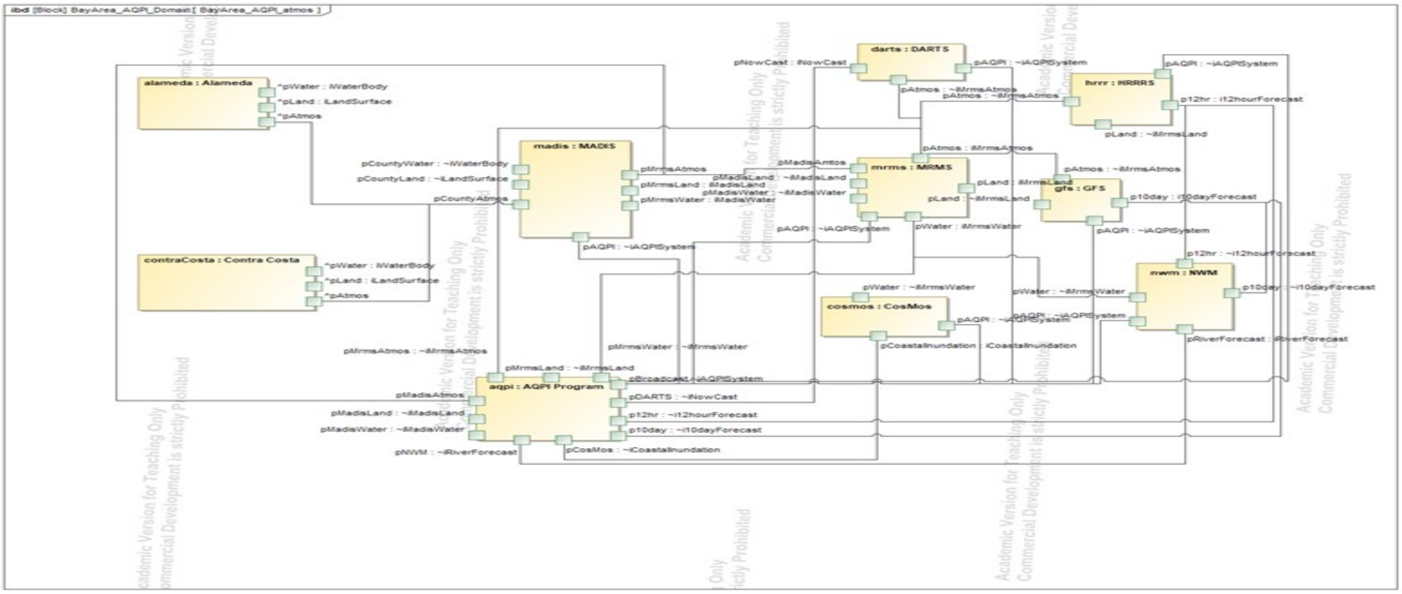

Fig. 4 AQPI System Internal Block Diagram

between blocks, which helps define the interfaces. All the exchanges are captured in an Internal Block Diagram (IBD); however, to keep a diagram readable, several IBDs can be used. Any component can have an IBD if there are internal workings that need to be captured and documented. For the AQPI system, only the atmospheric data and forecast exchanges are shown in Fig. 3.

C. *Behavior Modeling*

The model enables analysis of several aspects including internal processing, timeliness, and user interactions. Behavioral analysis of the User Interaction ensures that the AQPI provides a positive user experience, leading to the adoption of the tool into the users' weather analysis toolbox. To perform behavioral analysis, the model needs to include behavioral components, which include Use Cases, Activity Diagrams, and State Machines.

1) *Use Cases*

Use Cases capture the functionality of the system, identifying the actors (external) involved with each use case. Working with the Subject Matter Experts (SME), the systems engineer identifies multiple use cases to define the functionalities of the system to be developed. One Use Case of importance to user interaction behavioral analysis is Alerting for Bad Conditions (Fig. 4).

On the AQPI IBD, two counties are represented (Alameda and Contra Costa) which can provide three types of data: water body, land surface, and astmospheric. For this simulation, only the atmospheric data is modeled, indicated by the line connecting the pAtmos port on the counties to the pCountyAtmos port on MADIS. MADIS transmits the atmospheric data to both MRMS and the AQPI over the appropriate atmospheric data ports. MRMS transmits the atmospheric data to the different forecast models. The forecast models provide their forecasts to the AQPI program over different ports set to the appropriate interface definitions.

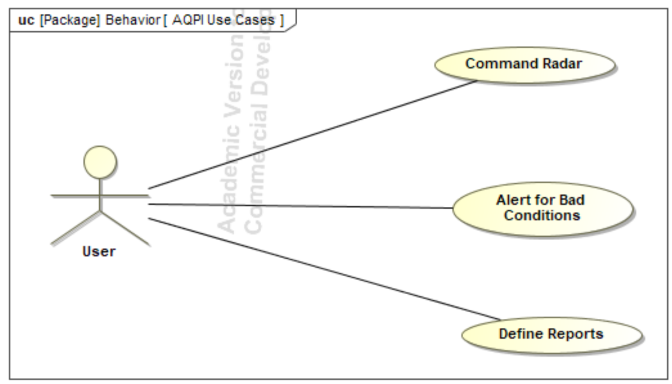

Fig. 3 AQPI Use Case Diagram

2) *Activity Diagrams*

A use case can be decomposed into Activities, Sequences, or both. For AQPI, Activities are chosen to capture the necessary actions to accomplish the use case. For the use case Alerting for Bad Conditions, the actions for each component of the AQPI are captured in Fig. 5. Several blocks perform actions, including the counties (represented by Alameda County), Meteorological

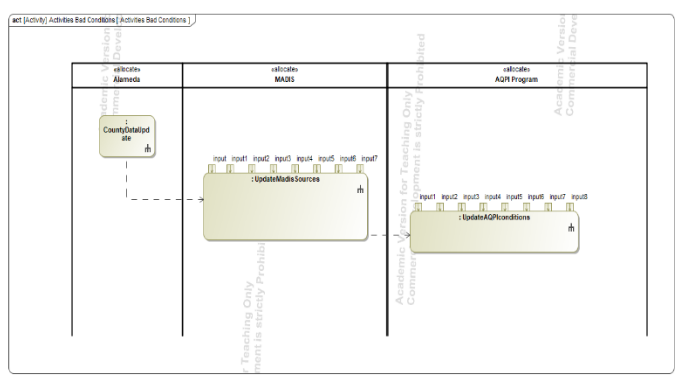

Fig. 5 AQPI Activities for Alert for Bad Conditions

Assimilation Data Ingest System (MADIS), and AQPI Program. Fig. 6 through Fig. 8 provide the details of the execution of the Alert for Bad Condition.

Almeda County provides updates for each of the types of sensor data (Fig. 6). Only Atmospheric updates are provided in this simulation; therefore, the Land and Water updates are only modeled with a timer and signal. For Atmospheric data, the Excel file containing the data is read for the next set of information. The temperature, dewpoint, station pressure, precitionation total, relative humidity, wind direction, wind speed, and wind gust populate the different inputs to the AtmosUpdate Signal, which is transmitted to the rest of the system.

MADIS receives the atmospheric data over the pCountyAtmos port (Fig. 7). Upon receipt of the signal, MADIS performs unit conversions as required and populates the signal MadisAtmosUpdates. This signal is transmitted to MRMS and the AQPI program over the pMrmsAtmos port. The APQI program process the received signal (Fig. 8), checking to see if any thresholds are violated and setting appropriate indicators.

3) State Machine

A State Machine describes the lifecycle of a block. Each block has a State Machine Diagram describing the states of that block, what causes the entry and exit (triggers) of a state, as well as what is done while in that state. Fig. 9 through Fig. 11 provide the state diagrams for the Alameda, MADIS, and AQPI Program blocks. Alameda begins updating sensor readings, indicated by the "do/AlamedaUpdate" in Fig. 9. "AtmosUpdate" signal provides the update from Alameda, triggering the transition to the state "Reporting" for MADIS (Fig. 10). Upon receipt of the sensor updates from Alameda county, MADIS updates and reports the conditions in the database through "do / updateMADIS". The AQPI Program receives the "MadisAtmosUpdate" and enters the monitoring state (Fig. 11). In the Monitoring state, AQPI Program evaluates the information by executing "do / newConditions".

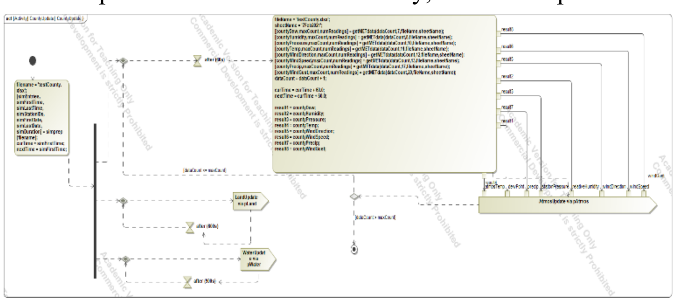

Fig. 6 CountyUpdate Activity Diagram

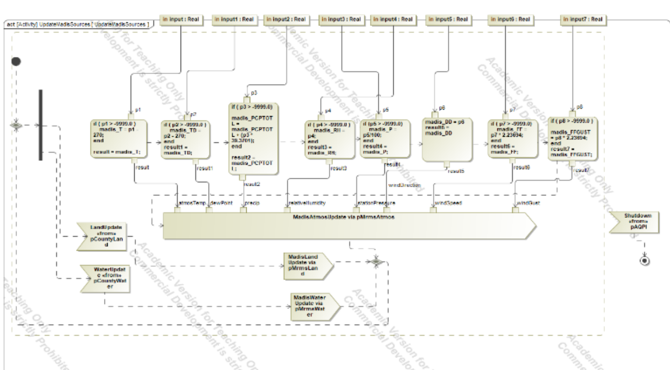

Fig. 7 UpdateMadisSources Activity Diagram

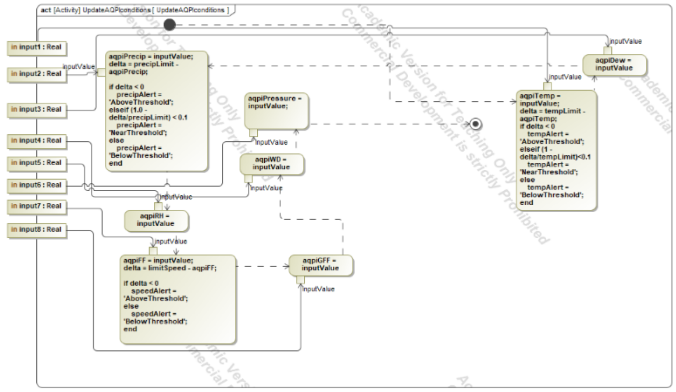

Fig. 8 UpdateAQPIConditions Activity Diagram

A state can, and most likely will, address more than one use case when the behavioral analysis complete. However, during the initial stages, a block's state may only apply to a single use case. Given the early stages of behavioral analysis for AQPI, the states are limited to single use cases.

D. User Interface Prototyping

Prototyping the display to monitor conditions and alert states enables the users and engineers to develop a concept Graphical User Interface (GUI) to be used as a template for the development team. Cameo Simulation Toolkit provides the capability to prototype GUIs. These prototypes, although rough, allow the users to interact with the simulation and provide

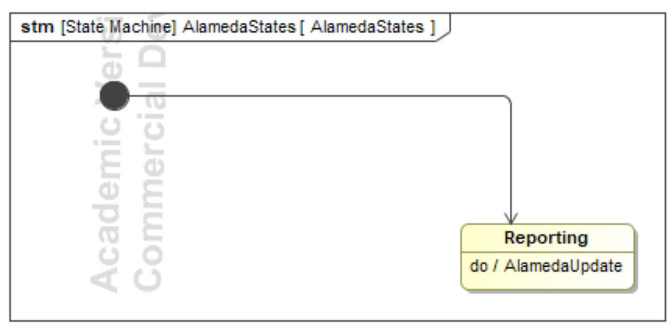

Fig. 9 Alameda State Machine Diagram

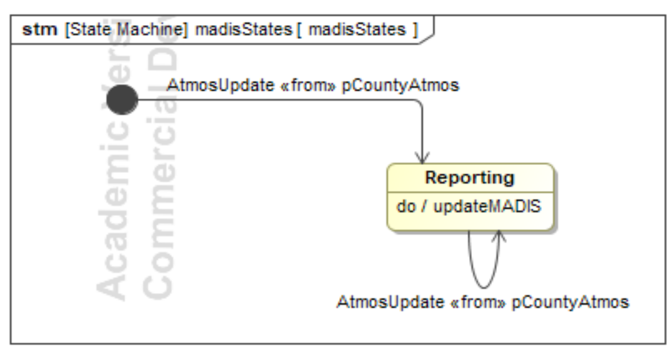

Fig. 10 MADIS State Machine Diagram

feedback on how the system should behave with user responses. The current prototype is shown in Fig. 12.

The prototype GUI has three panels: County Status, MADIS Status, and AQPI Commands. The County Status provides the status of the information coming out of the county without adjustments for units. MADIS Status displays the conditions being reported by MADIS, which have been converted to different units. The AQPI Command panel indicates the active thresholds along the top, while providing current conditions below. Status of atmospheric conditions with regards to the thresholds is provided by the color-coded indicator (green, yellow, or red) for below, within 10%, or exceeds threshold values.

To evaluate behavioral responses, the GUI is active during simulation execution allowing for the user to change the thresholds. By having this real-time interaction with the user, engineers observe both what the user is looking at as well as any extraneous actions which can be eliminated with an update to the prototype.

VI. Simulation

Cameo EA allows for dynamic simulation of the model using the Cameo Simulation Toolkit. Through both native simulation capabilities with Cameo, as well as the ability to call 3rd party tools (e.g., MATLAB), systems engineers can perform behavioral analysis on the system including sensitivity analysis, interface exchange validation, and user response.

The Bay Area AQPI domain simulation leverages data contained in Excel Spreadsheets and MATLAB models to enable realistic evaluations of the AQPI system. Starting with the county sensor data, the model pulls the atmospheric readings

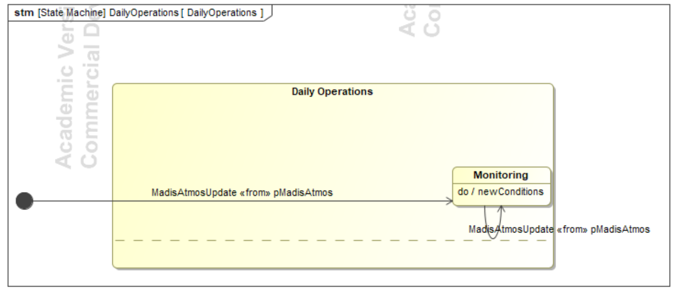

Fig. 11 AQPI Program State Machine

and transmits them to MADIS and MRMS. The forecast model elements receive the meteorological data from MRMS. Calling MATLAB functions, the forecast models provide forecasts to the system. The AQPI program receives these forecasts, along with the raw observation data, for alert processing and display to the user.

A simulation dashboard (Fig. 13) allows the systems engineer to track what is happening within the simulation and identify possible issues. The upper left diagram is the County Update activity (Fig. 6), with the Bay Area AQPI IBD (Fig. 3) is below. The MADIS update AD (Fig. 7) is positioned in the upper right. Below the MADIS update AD from left to right are the New Conditions AD and the AQPI update conditions AD (Fig. 8). For simulation execution purposes, the BayArea_AQPI_Domain block is also on the dashboard.

During simulation, the dashboard combined with the prototype GUIs enables real-time behavioral analysis. The dashboard provides a visual of what is happening using color

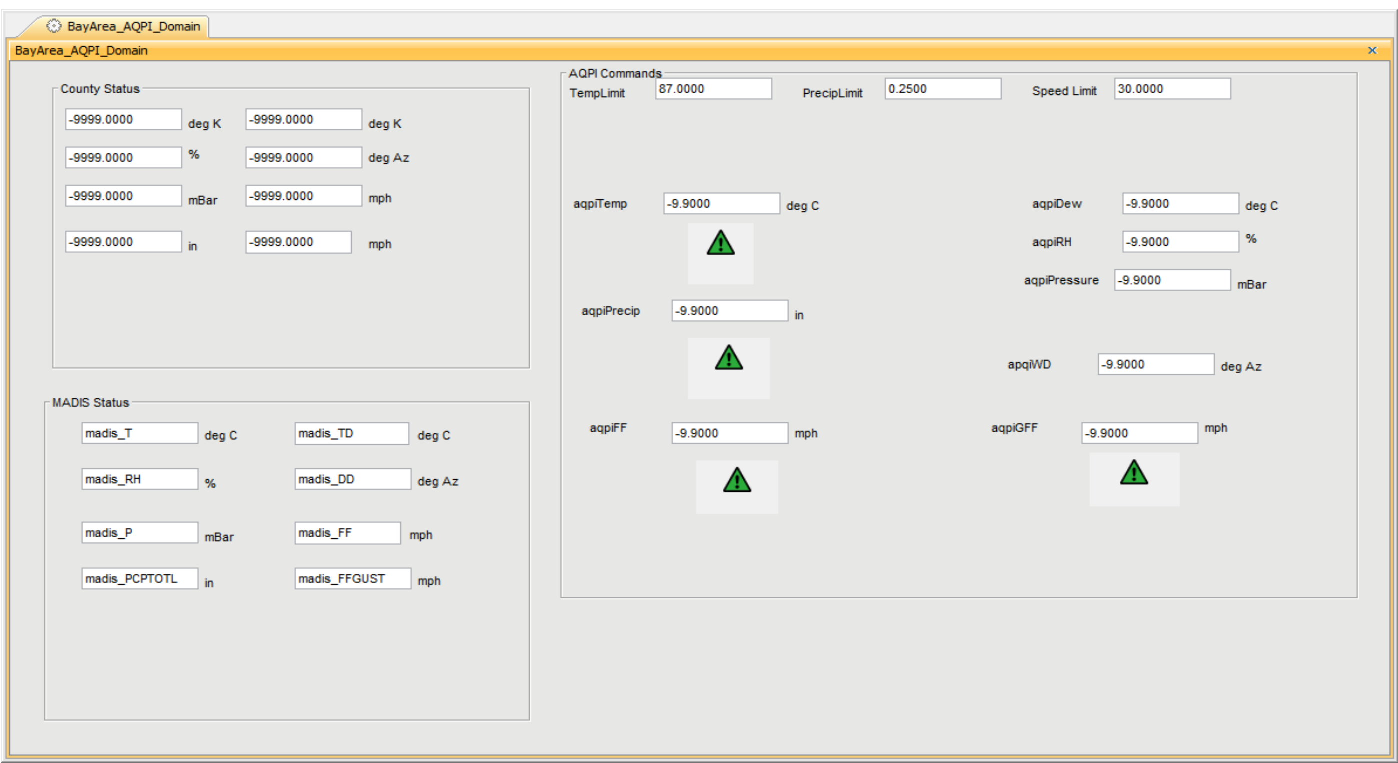

Fig. 12 AQPI GUI for Alert for Bad Conditions

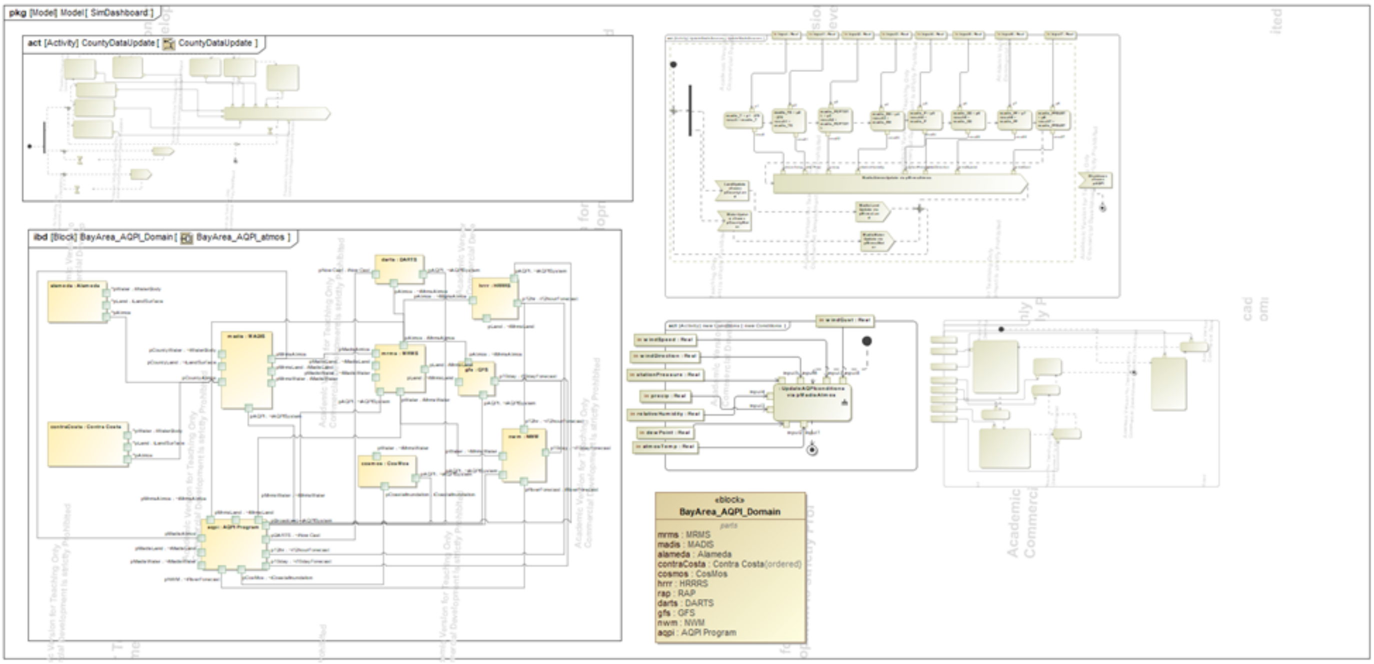

Fig. 13 AQPI Simulation Dashboard

codes as data are passed and actions are executed, while the GUIs provides a prototype user experience.

VII. Summary

By utilizing Modeling Based Systems Engineering (MBSE), AQPI systems designers can evaluate the design through dynamic simulation. Interface exchanges are identified through BDs and IBDs, while the Activity diagrams provide the understanding of the use of the data exchanges. Prototype GUIs employed during simulation execution provides behavioral responses. Behavioral analysis for each specific user group enables identification of the requirements necessary to satisfy the needs of the various users, develop the prototype GUIs, and perform sensitivity analysis based upon system conditions.

References

[1] NOAA Earth System Research Laboratory, "Project Description: Bay Area Advanced Quantitative Precipitation Information (AQPI) System," NOAA Earth System Research Laboratory, Boulder, CO, 2015.

[2] S. Friedenthal, A. Moore and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, Third Edition, Waltham, MA: Morgan Kaufmann, 2015.

[3] NOAA, "Physical Sciences Laboratory: Atmospheric River Portal," NOAA, [Online]. Available: https://www.psl.noaa.gov/arportal/about/. [Accessed 12 November 2020].

[4] F. M. Ralph, P. J. Neiman, G. A. Wick, S. I. Gutman, M. D. Dettinger, D. R. Cayan and A. B. White, "Flooding on California's Russian River: Role of atmospheric rivers," Geophysical Research Letters, vol. 33, no. L13801, 2006.

[5] J. Zhang, K. Howard, C. Langston, B. Kaney, Y. Qi, L. Tang, H. Grams, Y. Wang, S. Cocks, S. Martinaitis, A. Arthur, K. Cooper, J. Brogden and D. Kitzmiller, "Multi-Radar Multi-Sensor (MRMS) Quantitative Precipitation Estimation: Initial Operating Capabilities," Bulletin of the American Meteorological Society, pp. 621-637, 2016.

[6] AQPI Observations Team, "AQPI Initial Observations Report," NOAA Physical Sciences Laboratory, Boulder, CO, 2019.

[7] L. E. Johnson, R. Cifelli and A. White, "Benefits of an advanced quantitative precipitation information system," Journal of Flood Risk Management, vol. 13, no. S1, 2019.