Analysis of a Top-Down Bottom-Up Data Analysis Framework and Software Architecture Design

Anton Wirsch

Working Paper CISL# 2014-08

May 2014

Acknowledgement: Research reported in this publication was supported, in part, by the Charles Stark Draper Laboratory's University Research and Development program.
Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the Charles Stark Draper Laboratory.

Composite Information Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

Analysis of a Top-Down Bottom-up Data Analysis Framework and Software Architecture Design

by
Anton Wirsch

B.S. Electronics Engineering Technology (1998)
Brigham Young University

M.S. Computer Engineering (2004)
California State University, Long Beach

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management
at the
Massachusetts Institute of Technology
May 2014

Signature of Author: ____
Anton Wirsch
System Design and Management Program
May, 2014

Certified by: ____
Stuart Madnick
John Norris Maguire (1960) Professor of Information Technology,
MIT Sloan School of Management
& Professor of Engineering Systems, MIT School of Engineering

Approved by: ____
Patrick Hale
Director
System Design and Management Program

An Analysis of a Top-Down Bottom-up Framework and Proof of Concept Software Architecture
by

Anton Wirsch

Submitted to the System Design and Management Program in Partial Fulfillment of the
Requirements for the Degree of Master of Science in Engineering and Management

Abstract

Data analytics is currently a topic that is popular in academia and in industry. This is one form of bottom-up analysis, where insights are gained by analyzing data. System dynamics is the opposite, a top-down methodology, by gaining insight by analyzing the big picture. The merging of the two methodologies can possibly provide greater insight. What greater insight that can be gained is research that will be required in the future. The focus of this paper will be on the software connections for such a framework and how it can be automated. An analysis of the individual parts of the combined framework will be conducted along with current software tools that may be used. Lastly, a proposed software architecture design will be described.

Table of Content

1 Introduction

1.1 Motivation

In recent years the amount of data that is being generated by people and machines have greatly increased. Buzzwords such as Big Data, Internet of Things, and Machine-to-Machine Communication are commonly heard in mainstream media and indicate how prevalent the topic is. The potential benefit from vast amounts of data is that greater knowledge may be gained by analyzing the data. This type of analysis is a bottom-up approach and many organizations are implementing this approach. A top-down approach starts from general principles and works down to develop models of a process. This thesis investigates an architecture that combines the bottom-up approach with a top-down approach and reviews software tools that can realize the combined architecture.

1.2 Framework

A proposed framework of the combined methodologies has been provided, which will be discussed in detail in chapter 3. The framework consists of a top-down module and a bottom-up module along with connections between the two and other blocks. The framework will be analyzed to determine which portions of the proposed framework are applicable and which are not, as well as which portions are capable of automation. The resulting framework will then be used to design a software architecture that can be used to construct the framework.

1.3 Software Architecture and Tools

After the analysis of the top-down bottom-up framework, the resulting framework will then be used to design a software architecture. Existing data mining and system dynamics tools will be leveraged to propose a software implantation of the software architecture. The feature set and automation capabilities of data mining and system dynamics tools will be analyzed to determine which of the tools are applicable to the software implementation.

1.4 System Dynamics and Data Mining

System dynamics and data mining are implementations of top-down and bottom-up approaches respectively. Both are heavily used in business. One example of data mining in business is determining which subset of potential customers to advertise to. A company can analyze their database of customers to determine which types of people are the most common. Knowing this the company can target those types of people for advertisement instead of covering all types. System dynamics is often used to model the policies of a corporation. A simple example will be modifying the inventory policy of a corporation. Various inventory policies can be simulated to see how the change will effect inventory and the overall supply chain over a set period of time. A system dynamics model can be packaged as a “flight simulator” to allow managers to experiment with adjusting parameters and policies and seeing how the system behaves.

The operational methods of the two systems differ. Data mining is used in a live setting where new data is processed on a continuous basis. It is also usually highly automated where there is little to no human interaction required to operate the data mining system. The main use case for system dynamics on the other hand, is for an interactive simulation test environment. A user can set various parameters of the model and then execute a simulation to produce a time-series output. The combined framework and resulting software architecture will be the combination of the two. The framework will operate as an automated system, conduct simulations, and produce a time-series output at a predetermined time interval.

1.5 Purpose

While data mining and system dynamics are used in business the combined framework as described here, will not be used for business use. Instead the use case will be to monitor and forecast various events that occur throughout the world. Another use case is to analyze historical events to help understand the important factors of the event. Riots are an example of an event. They occur frequently throughout the world and cause significant damage to a city such as the 2011 England riots.

The goal in this example is to see if the combined framework can forecast a riot. This will allow authorities to allocate resources and take action to help prevent the riot or prepare for the riot. Years of research will be required to test this theory. This thesis will provide the framework and software architecture to enable the start of the research.

1.6 Summary of Chapters

Chapter 2 will provide an overview of data mining and system dynamics. It will cover their strengths and limitations as well as the process to implement both methods.

Chapter 3 will analyze a framework that incorporates top-down and bottom-up methods. It will go over the various parts of the framework and any modifications that were made.

Chapter 4 will explore the software tools that are available data mining and system dynamics. A number of commercial and open source tools will be analyzed for their feature set for use in the software architecture.

Chapter 5 will discuss the software architecture and data mining and system dynamics tools that can be used for the construction of the software architecture.

Chapter 6 will provide a summary of the thesis.

2 Top-Down Bottom-up Overview

2.1 Bottom-up

2.1.1 Overview

Bottom-up (BU) analysis consists of analyzing various forms of data, such as numbers, text, images, video, voice, etc. in order to find relationships and patterns to gain knowledge from the data. Bottom-up analysis has experienced tremendous growth over the years. This type of analysis has spread to a number of sectors including finance, business, law enforcement, and defense to name a few. Data mining, machine learning, and big data are commonly representatives of bottom-up analysis. This thesis will focus on the use of data mining when referring to bottom-up analysis.

2.1.2 Data Mining, Machine Learning

“Data mining is the process of exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules.”[1]

The above quote provides a simple explanation to data mining. It is a system where data is gathered, stored, and then analyzed in an automated method.

One business case example of data mining is to determine if a person will apply for a credit card if provided an advertisement for a card. To understand what types of people are likely to apply for a credit card, credit card companies store attributes of each person that has joined. The attributes can include age, gender, occupation, income, marital status, home address, etc. A number of analytical methods can be used to determine what combination of attributes that a person has will likely apply for their credit card. This analysis portion is where machine learning is implemented. Past data is used to help create an algorithm that learns what combinations provide the highest probability that a person will apply for a credit card. The learned algorithm is called a fit algorithm. Once the fit algorithm has been developed it is used within a data mining system for live operation. Returning to the example, instead of mailing millions of random people credit card applications, a data mining system can parse through a database of noncredit card holders and test the attributes of each person against the fit algorithm. Those that are determined to likely apply for credit cards can be the recipients of a credit card application.

Machine learning is a method for learning from data in an automated fashion. The task of determining if an email is spam or not will be used as an example to explain machine learning further. A program or model, which contains a number of parameters, can learn if an email is spam or not through repeated exposure to spam email and nonspam email. Each exposure will adjust parameters to improve performance. This is an automate process where model itself is adjusting parameter to improve performance. Once the performance is at an acceptable level the model is considered fit. This thesis will refer to this fit model as the data mining model.

The main algorithms used for machine learning are classification, clustering, regression or prediction, and association rule. For classification, the goal is to classify something to a predetermined set of categories. For example if a person is provided an advertisement for a credit card, will the person be likely to sign up or not. Only two possibilities exist for this case.

Clustering will group data into similar categories where the number of categories has not been predetermined. Data points, often referred to as records, that are similar are grouped together. A business example of clustering is comparing the amount of income and debt of a set people. Points that are close to each other will be clustered. The figure below shows a plot of the points and the resulting three clusters.

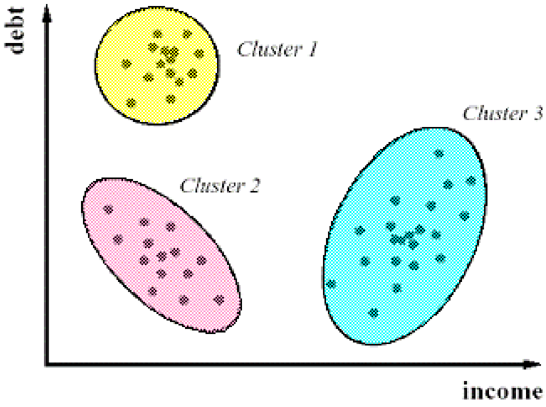


Figure 1 Cluster Example [2]

Regression will predict a value, such as the price of a house depending on attributes such as the age of the house, number of rooms, neighborhood, etc. By analyzing the price of houses along with their attributes that have sold over the years a model can be created. The model can then be used to predict what the price of a house would sell for based on the attributes of the house.

Lastly, association rule determines what objects are usually associated with each other. Super markets are interested in this type of data. They are interested in knowing what other items are usually purchased with hotdogs.

These four categories fall into two general groups supervised learning and unsupervised learning. Supervised learning, which includes classification and regression algorithms provide feedback. If a classification of a record is correct or incorrect the feedback can be used for learning. Unsupervised learning, which includes clustering and association rule do not provide any feedback. Therefore learning cannot be gained by processing historical data. For example, clustering will group similar data points but since there is no predetermined number of classification to fall into there is no feedback to learn if the clusters are correct or not.

Each of the four categories can be implemented through a number of algorithms. For example classification is possible through decision trees, neural nets, Bayesian classifiers, and Support Vector Machines to list just a few. When a classification machine learning model is being constructed a number of algorithms will be tested to see which will perform the best. The same process is conducted with the other categories as well.

2.1.3 Data Mining Flow

The flow for data mining is described below.

Step1: Develop an understanding of the purpose of the data mining project. Is the purpose a one time effort or will it deal with executing countless times.

Step 2: Obtain the dataset to be used in the analysis. If the amount of data is extremely large then it may sufficient to randomly sample a portion of the data. A thousand records is usually enough for creating a model [3]. Data may need to be queried from multiple databases internally and externally.

Step 3: Explore, clean, and preprocess the data. Data may be plentiful but often it is not clean. Missing records from a dataset is common. A decision must be made on how to handle missing data. It can be ignored or averaged between the surrounding records. Incorrect data is also common. For this case obviously incorrect data can be checked for. For example if the expected value of a record is between two values and the record is outside the range then this record holds incorrect data an can be ignored.

Step 4: Reduce and separate the variables. Not all variables may be needed. At this step variables that are not required are removed from analysis. The more variables are included the more CPU time will be required for processing. Therefore it is ideal to keep the number of variables as low as possible. For example, assume that a house has 20 attributes. If all 20 attributes are used to create a fit model then the fit model will have to use all 20 attributes for each record it processes. If the same or slightly less accurate fit model can be created with only 5 attributes then the CPU load will be considerably less.

Also, some variables will need to be modified or transformed. For example if a variable is the age of a person the resolution may be too fine. It may be easier to analyze if a number of age ranges were used instead.

Lastly when supervised training will be used the data should be split into three groups training, validation, and test. The training set is used to train a model. Once it is trained the validation and test set will be used to see how it performs with a different set of data.

Step 5: Choose the data mining task (regression, clustering).

Step 6: Use algorithms to perform the task. This will usually take many attempts. Various combinations of variables as well as multiple variants of the same algorithm will be tested. Promising algorithms can be tested with the validation dataset to see how it performs against a fresh set of data.

Step 7: Interpret the results of the algorithms. An algorithm from one of the many tested in step 6 needs to be chosen. The chosen algorithm should also be tested against the test dataset to see how it performs with yet another set of new data. At this point the algorithm has been fitted for the task at hand.

Step 8: The fit algorithm is integrated to the system for use with real data. The system will execute the fit algorithm against the new records to make a determination such as what classification or clustering does the record belong to or what is the resulting numerical value, or what is the record associated with. Appropriate action for each possible outcome must then be taken.

2.2 Top-down

2.2.1 Overview

While bottom-up analyzes data to uncover patterns, a top-down method approaches a problem from the high level view a system. For example, in a bottom-up business example a company will look at sales and customer data to extract any patterns. A top-down approach could model a corporation and its strategy. Who are the target customers? What is the supply chain? What is the marketing? How are the departments divided? What are the corporate sales policies? How are sales team incentivized? There are external factors that also need to be considered in this example such as competition from other companies and the overall economy. If all the divisions of the company are not aligned with the corporate strategy then it will be easy to understand that the target sales and customer reach will not be optimal. By starting from the top and then deconstructing the parts and understanding the interactions between the parts one can gain an understanding of what type of customers can be reached and attracted. This thesis will focus on system dynamics (SD) as the implementation of top-down analysis.

2.2.2 System Dynamics

Jay Forrester, who was a professor in the school of management at MIT, developed system dynamics in the 1950s [4]. System dynamics models complex systems and observes the behavior of the model over a period of time. The observations are conducting by executing simulations of the model and visually viewing the output through graphs and charts. Complex systems, where the aggregate relationships among multiple nodes are difficult to understand, are the type of systems that are suited for system dynamics. The strength of system dynamics is discovering relationships within the model that are not obvious. Simulations that are run on the model show how the relationships among nodes affect each other over a period of time [5].

One of the first implementation of SD was with one group in GE. They observed wild fluctuations on orders of a particular household appliance. At one point they may be backlogged on large orders and at other times there would be little orders with much of the staff having nothing to do. Forrester was asked to solve the problem and he did so using system dynamics. He found that by modeling the management policy there was a long delay from when retailers put in their orders to when they would receive the product. Retailers anxious for their orders to arrive would pile on more orders. As the order went through the supply chain each step in the chain anticipated an increase in demand and would add more to the order, which eventually became increasingly inflated by the time it reached GE. As the retailers received their orders their supply soon surpassed demand leading to no orders, which would then again propagate through the supply channel. This was the basis of the cyclical orders [6].

The above example shows how system dynamics can be used to explain puzzling issues of a corporation. In this case the issue was not with one particular department but with the overall ordering system. Also note that the retailer and throughout the supply chain the actual order was not placed but additional orders were added. Orders were inflated because higher orders were anticipated. Enough people in the whole chain held the same belief and acted on it, which contributed to the phenomenon. These beliefs cannot be measured directly, but certainly affected the system. The beliefs that individuals hold in aggregate across a population are referred to as narratives. System dynamics is also able to include these unmeasurable concepts within a model. This capability separates system dynamics from data mining and other forms of bottom-up methodologies.

2.2.3 System Dynamics Model Creation Method

There are a number of current prominent thinkers in SD among them are Randers, Richardson and Pugh, Roberts et al., Wolstenholme, and Sterman. Each has defined a flow for creating a SD model.

Randers (1980)	**Richardson and Pugh (1981)**	**Roberts et al (1983)**	**Wolstenholme (1990)**	**Sterman (2000)**
Conceptualization	Problem definition	Problem definition	Diagram construction and analysis	Problem articulation
	System conceptualization	System conceptualization		Dynamic hypothesis
Formulation	Model formulation	Model representation	Simulation phase (step 1)	Formulation
Testing	Analysis of model behavior	Model behavior		Testing
	Model evaluation	Model evaluation		
Implementation	Policy analysis	Policy analysis and model use	Simulation phase (step 2)	Policy formulation and evaluation
	Model use			

Table 1 Prominent System Dynamics Leaders' Modeling Processes [7]

Further details of the process proposed by Sterman are explained below.

Problem Articulation

The first step in designing a model is to set the bounds. What is the problem or question that needs to be solved? The bounds should be set enough to model the problem. What should be avoided is setting no bounds and attempt to model an all-encompassing system. This will lead to a complex model that may never properly function correctly [8].

Dynamic Hypothesis

At this stage hypotheses for the problem can be entertained. What variables will be required for the hypothesis? Variables fall into three categories: endogenous, exogenous, and excluded [9]. Endogenous variables are variables that are within the boundary of the problem. Exogenous variables are variables that are outside of the boundary. In general there should be very few exogenous variables used. Lastly, excluded variables are variables that are not used since they have little to no affect the outcome of a model.

Formulation

This state entails building a full model and submodels with all connections and equations.

Testing

Testing and Formulation are an iterative process. The first test is to verify that the behavior of the model is matching with expected behavior. Many iterations of testing and modifying the model is usually required before the model behaves properly.

Another aspect of testing is ensuring units consistency. As the model is being constructed, each node will have an equation associated with it. Units in the equations should match. If a stock has three other nodes as inputs then the three nodes should have the same units. Software tools will usually have a feature to check for unit consistency.

Sensitivity analysis is another check that needs to be done. The purpose is to see how sensitive the model is to changes to the input. Monte Carlo is the usual method to test for sensitivity and is again a feature that is regularly included in system dynamics software tools.

Sterman stated that an important test that is often missed is reality tests. These are done by testing extreme cases even though the cases will never occur. The purpose is to ensure that the model is following basic reality and by testing extreme conditions the outcome should be obvious [10]. For example if an infectious disease model was created and the population was set to zero then the infected population should be zero, not a positive or negative value.

Policy formulation and evaluation

After testing has completed the model is ready to be deployed. In many cases system dynamics models are used as an environment to test ideas and scenarios. This type of usage is referred to as a flight simulator. In the business field different operational policies can be tested and evaluated before any are actually implemented.

2.2.4 Model Components

System Dynamics models are composed of stocks, flows, convertors, connectors, sources, and sinks. Each component will be briefly explained. The below figure is a simple rabbit population model. The components of the model are indicated in red text. The example model will be referenced in each component explanation.

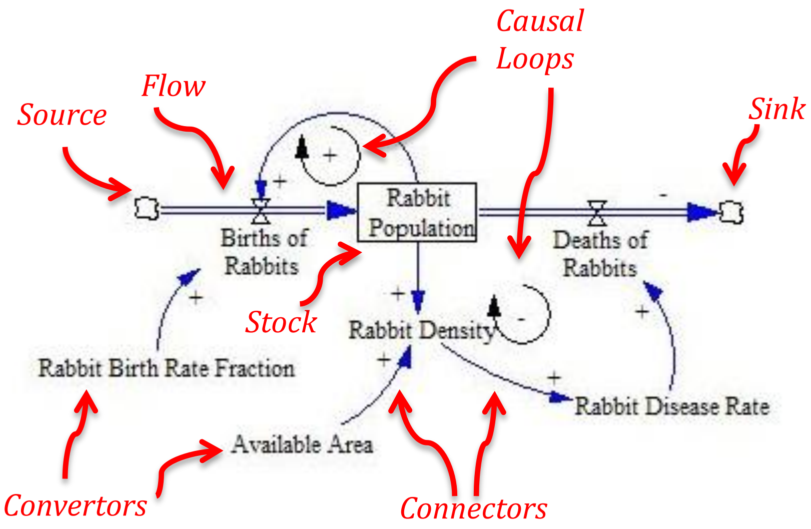


Figure 2 Rabbit Population Model [11]

Stocks

Stocks represent objects or ideas that can build up. A bathtub is a common example. Water can be added to the bathtub and the level of water indicates the stock of water. In the example model the Rabbit Population is stock. The rabbit population can increase or decease.

Flows

Flows are rates that affect the inflow or outflow of stocks. Using the bathtub example again, the rate at which the bathtub fills will be a flow and the rate at which the bathtub drains will be another flow. In the example model, Birth of Rabbits is an inflow and Deaths of Rabbits is an outflow.

Convertors

Convertors are variables that are neither stocks nor flows. They can be variables that are at the edge of the system boundary and only have an output. They can also be intermediate variables. In the example model Rabbit Birth Rate Fraction and Available Area and Rabbit Density are convertors. Notice that Rabbit Birth Rate Fraction and Available Area only have outputs. These are the boundary variables. Rabbit Density is an intermediate variable since it has inputs and an output.

Connectors

Connectors connect convertors to other convertors or are connected from stocks or flows.

Source and Sinks

Source and sinks are stocks that are outside the boundary of the model. In the example model Births of Rabbits comes from a source and Deaths of Rabbits goes to a sink.

Causal Loops

As explained previously, system dynamics models are made up of six components. The connectors and flows define the relationships between the components. These relationships will contain loops. There are two types of loops, positive or reinforcing loops and negative or balancing loops. In the example model the Birth Rate of Rabbits will increase the stock of rabbits. The increase in Rabbit Population will then increase the Birth Rate of Rabbits. Both are reinforcing each other. On the other end the greater the Rabbit Population the greater the Density of Rabbits, which will increase the Disease Rate among rabbits. This will increase the Death Rate of Rabbits, which will decrease the Rabbit Population. This is a balancing loop since the greater the population the larger the rate of rabbit deaths will occur.

2.2.5 Information on Creating Models

Model creation is not a straightforward process. According to Sterman a mixture of quantitative and qualitative data is required to create a model [12]. Qualitative data often exist as a mental model of the modeler. Converting qualitative data to a quantifiable expression requires experience. When the modeler is not the domain expert and does not hold the qualitative data, domain experts will be required. With the inclusion of qualitative data, model creation becomes a nontrivial exercise. Expertise is required. However once a model has been created, using it and experimenting with values can be done by anyone with some tutorial.

2.2.6 Time

One difference between data mining and system dynamic models is that system dynamic models all have a time aspect. Simulations are executed with a predetermined time step. The time step is repeated until a predetermined number of time steps have passed. The simulation will result in a time-series output. The stocks in the model will usually have a different value from the start of the simulations to the end of the simulation. In the example model the rabbit population will change over time. Data mining can deal with time series data, but the majority of data mining model have no relations to continuous time.

3 Top-Down Bottom-Up Framework Analysis

3.1 Overview

To create a software architecture that will implement top-down and bottom-up methodologies a framework of how the system is connected is first required. Instead of creating a framework from scratch a proposed framework produced by Allen Moulton, a research scientist at the MIT Sloan School of Management, will be analyzed. Once the analysis is completed and modifications to the framework made the software architecture can then be designed.

For this analysis, the end goal is to create an automated framework in which the system is able to execute with minimal intervention by the operator. The below figure is the proposed base framework to be analyzed. Each part of the framework will be analyzed for feasibility and automation. The framework consists of two main modules, the top-down module in the upper right and the bottom-up module in the lower left. The top-down module will consist of a system dynamics model. This will execute a simulation and produce a time-series output. The bottom-up module will consist of a data mining model to provide input to the system dynamics model.

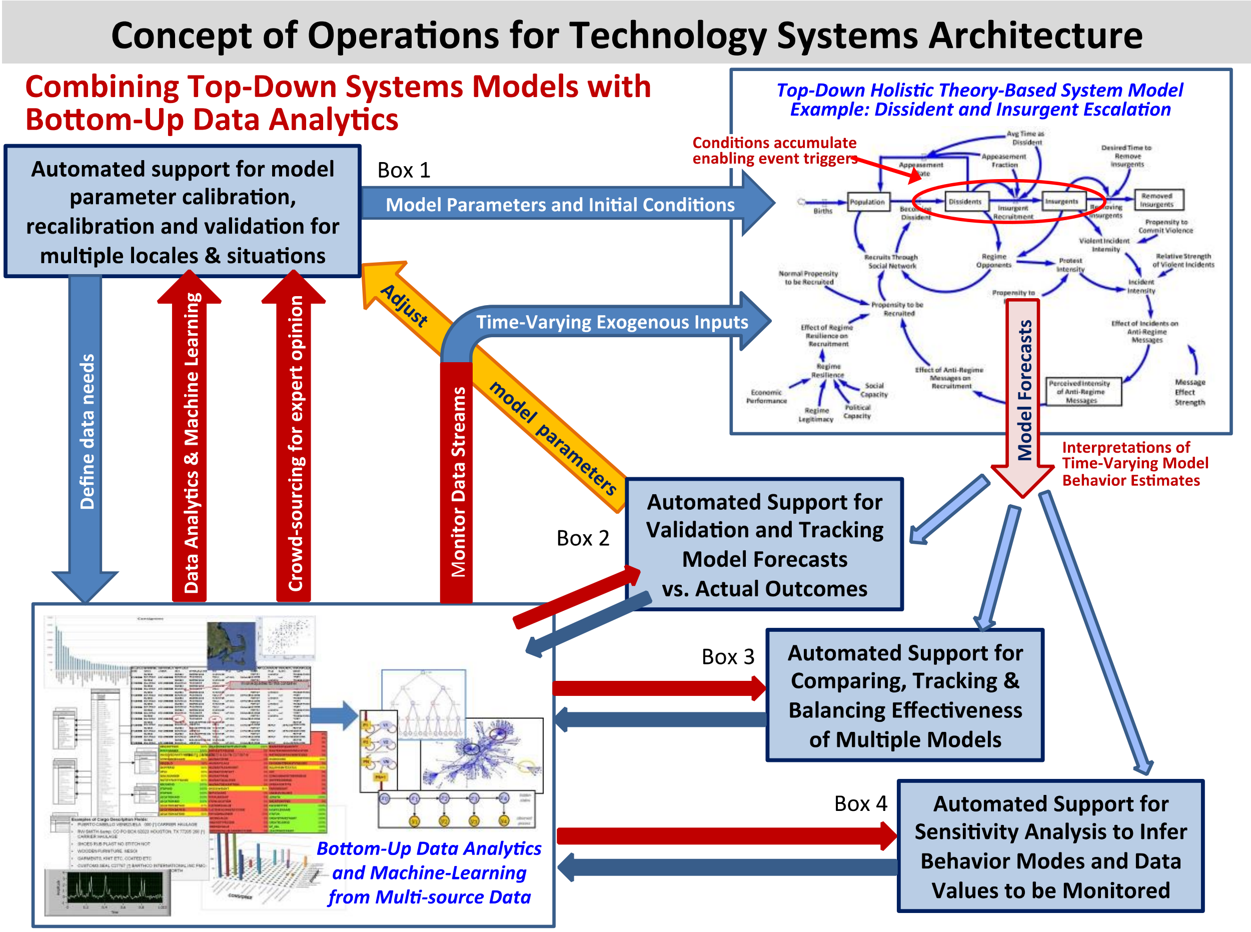

Concept of Operations for Technology Systems Architecture
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Automated support for model parameter calibration, recalibration and validation for multiple locales & situations
Box 1
Model Parameters and Initial Conditions
Top-Down Holistic Theory-Based System Model
Example: Dissident and Insurgent Escalation
Conditions accumulate enabling event triggers
Avg Time as Dissident
Appeasement Rate
Appeasement Fraction
Desired Time to Remove Insurgents
Births
Population
Becoming Dissident
Dissidents
Insurgent Recruitment
Insurgents
Removing Insurgents
Removed Insurgents
Propensity to Commit Violence
Violent Incident Intensity
Relative Strength of Violent Incidents
Recruits Through Social Network
Regime Opponents
Protest Intensity
Incident Intensity
Normal Propensity to be Recruited
Propensity to be Recruited
Propensity to
Effect of Regime Resilience on Recruitment
Effect of Incidents on Anti-Regime Messages
Regime Resilience
Effect of Anti-Regime Messages on Recruitment
Economic Performance
Social Capacity
Regime Legitimacy
Political Capacity
Perceived Intensity of Anti-Regime Messages
Message Effect Strength
Model Forecasts
Interpretations of Time-Varying Model Behavior Estimates
Adjust
Time-Varying Exogenous Inputs
model parameters
Define data needs
Data Analytics & Machine Learning
Crowd-sourcing for expert opinion
Monitor Data Streams
Box 2
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes
Box 3
Automated Support for Comparing, Tracking & Balancing Effectiveness of Multiple Models
Box 4
Automated Support for Sensitivity Analysis to Infer Behavior Modes and Data Values to be Monitored
Bottom-Up Data Analytics and Machine-Learning from Multi-source Data

Figure 3 Proposed Top-Down Bottom-Up Framework

3.2 England Riots

An example that will be used throughout the section is the 2011 England riots in which a series of riots erupted throughout England over a period of five days. The origin of the riots is with the fatal shooting of Mark Duggan, who was unarmed, by police. An accurate account of the shooting is still elusive, as facts of the incident have changed over time. A special police unit investigating gun crime in a black community stopped a mini cab that Duggan was in. Initial reports indicated that Duggan had exited the cab with a gun in hand, but later it was known that Duggan was unarmed when police fired upon him. The killing of an unarmed man brought protesters to the streets of Tottenham on August 6. The protest later developed into a riot where looting and arson occurred. Other riots emerged throughout England for the following five days [13].

3.3 Forecasting

One of the functions of the framework is to monitor an ongoing event such as protests that can lead to riots. This functionality will be referred to as the monitoring system. The proposed monitoring system has a different use case than a typical system dynamics model. Although it is possible to simulate at a yearly or greater interval, this monitoring system is focused to simulate for shorter time periods such as days, weeks, or possibly months. The purpose of the monitoring system is to monitor a currently evolving event such as protests or riots, which has a timescale from hours to days. As the event under monitor continues, data regarding the event will be gathered and become an input to the monitoring system, which can then execute a simulation to generate a forecast of how the current event will progress. This process will be repeated at intervals until the event had completed.

This system is analogous to hurricane weather forecasting. Climatologists gather atmospheric data and conduct simulations to predict the path of a hurricane based on the theory of how hurricanes develop. The forecast is reliable for the near future but as forecast goes beyond the near future the forecast becomes less reliable as shown in the below figure. As time progresses, they are able to collect further data and continuously update the forecast. The goal of the monitoring framework is to do the same for events. Once an event has been identified data regarding the event will be gathered and forecasts will be generated and regularly updated in an automated fashion.

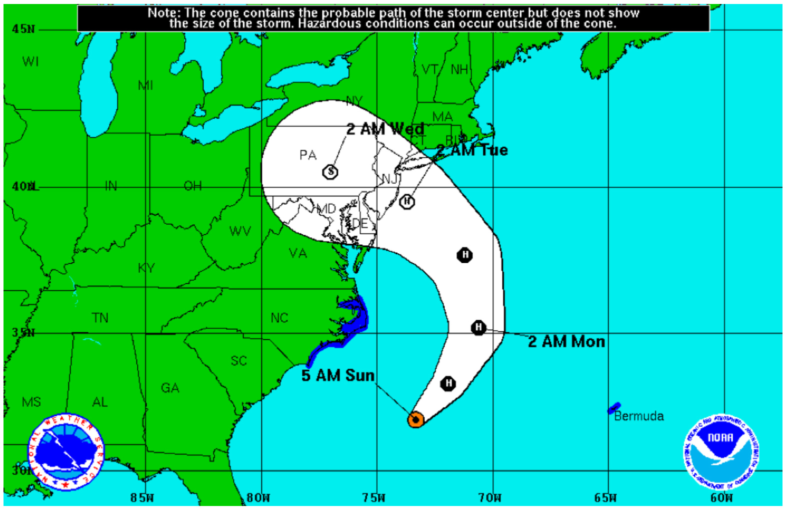

Note: The cone contains the probable path of the storm center but does not show the size of the storm. Hazardous conditions can occur outside of the cone.
45N
40N
35N
30N
85W
80W
75W
70W
65W
60W
WI
MI
IN
OH
KY
TN
MS
AL
GA
SC
NC
VA
WV
PA
NY
VT
NH
MA
CT
RI
NJ
MD
DE
2 AM Wed
2 AM Tue
2 AM Mon
5 AM Sun
S
H
H
H
H
Bermuda
NOAA

Figure 4 Hurricane Forecast [14]

3.4 Framework Uses

As stated previously one of the functions of the framework is to monitor an ongoing event. The purpose of the monitoring framework is to produce a forecast so that authorities can best allocate resources for actions that can reduce the possibility of an event worsening such as a protest evolving to a riot, or to prepare for the forecasted worsening event such as a riot. An additional use case is for simulating historical events. There are a number of uses for using historical events. One can be training sessions for actual events. Mock events can be created and be executed as if the event was occurring in real time. The second is to further improve the system and develop best practices.

3.5 Bottom-up Data Sources

Any data source that is accessible can be used for the bottom-up module. Social media sites such as Twitter, Facebook, and Google Plus are sources where the general public generates data. Of particular interest is Twitter, which allows users to post images or text message that are 140 characters or less. The users are any individual or organization that have created an account. Twitter is often the fist site where news is reported. Frequently news is posted to Twitter by individuals who are at the scene of an event. As of January 1 2014 Twitter has 645,750,000 active registered accounts [15]. The large user base provides a favorable probability that an event of interest will be covered through Twitter. The data that can be gathered from Twitter a user posts vary include posted message, location, time of post, author, author followers, and author prominence (how many accounts are following the author). Information that can be gained from the posted message can vary from personal opinion, location, keyword, ect.

During and before the England riots erupted a large volume of Twitter posts were generated. The posts included user's narrative of the government, for or against. Gathering these types of posts over a period of time allows monitoring the growth of particular narratives. User location is also useful. If a number of posts are coming from a particular location then it indicates that a particular event may be occurring at the location.

Another database that can be referenced is GDELT, which is an open database of events that occur throughout the world and covers over 300 categories staring [16]. The database covers over three decades starting from 1979 and is updated daily.

A formal description of GDELT is described in the following quote. “The Global Database of Events, Language, and Tone (GDELT) is an initiative to construct a catalog of human societal-scale behavior and beliefs across all countries of the world, connecting every person, organization, location, count, theme, news source, and event across the planet into a single massive network that captures what's happening around the world, what its context is and who's involved, and how the world is feeling about it, every single day.” [17]

GDELT records the event type, dates, locations, and prominent actors of an event. The resolution of the location may vary from country level down to the city level. The actors will not be an individual name but rather a classification such as resident, farmer, company, or even country names. The database include events such as protests, riots, military or police power, etc. GDELT has provided the complete database so that it can be downloaded and installed to a relational database. They also provide example SQL scripts to create the database tables and to import the data into a relational database.

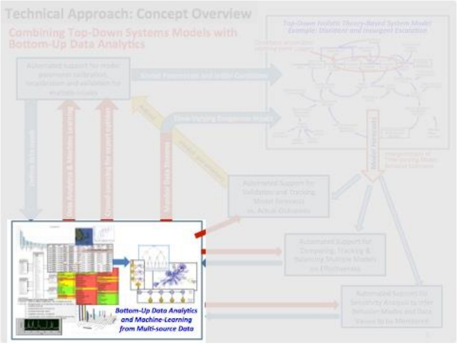
3.6 Multiple Models

As stated previously the framework consists of a top-down module, which is implemented using a system dynamics model. In actuality the framework is able to support multiple top-down implementations and instantiations. For example two different system dynamics models may be placed in the top-down module. The modeler may have multiple ideas on how to model an event. Instead of having to decide on one method multiple models can be created to see which performs best. System dynamics is one implementation of a top-down module. Other implementations methods are also possible such as agent-based models. For each top-down instantiation a matching bottom-up instantiation will be required.

3.7 Description of Monitoring Framework

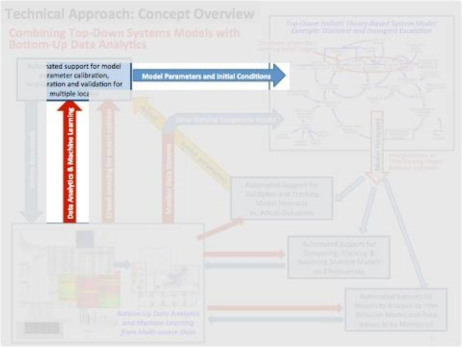
A brief explanation of each part of the proposed framework of figure 2 is provided below. To assist in identifying the portion under explanation a miniature version of figure 2 is provided with the portion under explanation highlighted. Further details of the framework will be discussed in the analysis section.

Bottom-Up Module



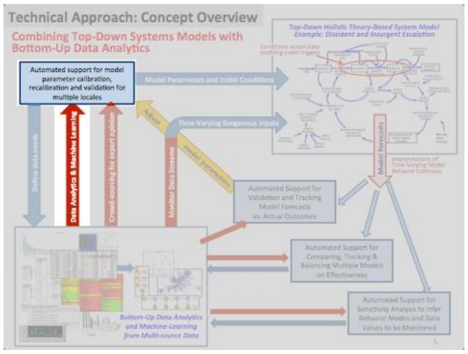
The bottom left box represents the bottom-up module. This box will gather data, clean, and process data. A data mining model will be at the core of the module.

Bottom-Up Outputs


Technical Approach: Concept Overview
Combining Top-Down Systems Models with
Bottom-Up Data Analytics
ated support for model
rameter calibration,
ration and validation for
multiple loca
Model Parameters and Initial Conditions
Data Analytics & Machine Learning

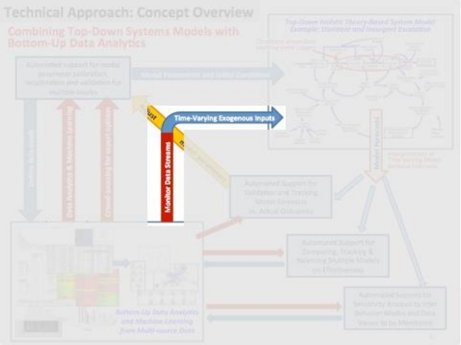
As data is gathered it will be processed through the model in the data mining module. The output of the processed data will then be passed to the top down module.

Crowd Sourced Expert Opinion


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Automated support for model parameter calibration, recalibration and validation for multiple locales
Model Parameters and Initial Conditions
Top-Down Holistic Theory-Based System Model Example: Dissident and Insurgent Escalation
Conditions accumulate enabling event triggers
Time-Varying Exogenous Inputs
Adjust
model parameters
Define data needs
Data Analytics & Machine Learning
Crowd-sourcing for expert opinion
Monitor Data Streams
Model Forecasts
Interpretations of Time-Varying Model Behavior Estimates
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes
Automated Support for Comparing, Tracking & Balancing Multiple Models on Effectiveness
Automated Support for Sensitivity Analysis to Infer Behavior Modes and Data Values to be Monitored
Bottom-Up Data Analytics and Machine-Learning from Multi-source Data
5

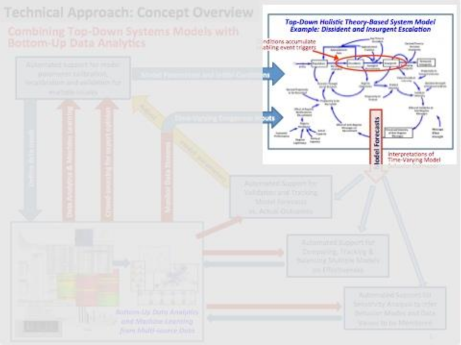
When monitoring an event it is advantageous to be able gain the opinion of experts in the domain. This output represents expert opinion that has been gathered by some means.

Monitor Data Streams


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Time-Varying Exogenous Inputs
Monitor Data Streams

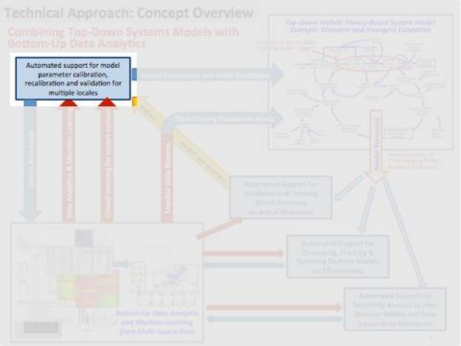
This is data that passes through the data analysis block. The data is cleaned, formatted and passed to the top-down block for its input. This data is not processed through the data mining model.

Top-Down Module


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Top-Down Holistic Theory-Based System Model Example: Dissident and Insurgent Escalation
nditions accumulate
abling event triggers
Model Forecasts
Interpretations of Time-Varying Model

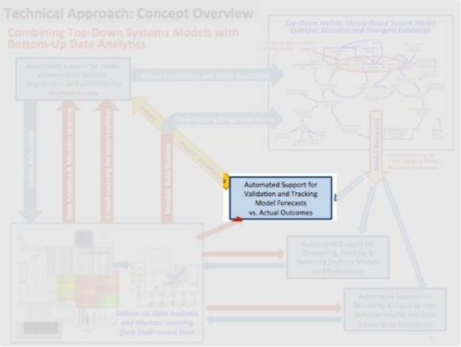
The upper right box is the top-down module. System dynamics models will be the main implementation method for this module. This module will execute a simulation and generate a time-series output.

Automated Support for Model Parameter Calibration (Box 1)


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Automated support for model parameter calibration, recalibration and validation for multiple locales

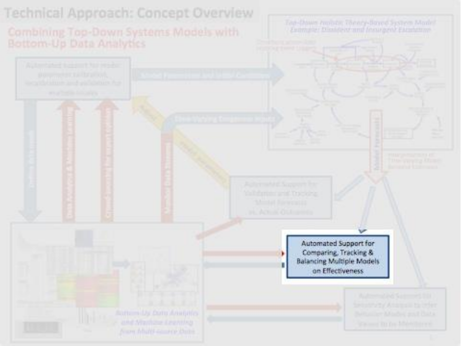
System dynamic models require calibration of parameters. When a data mining or system dynamic model is created many times they are created to a specific locale. For example a model of a riot in England may not have the same dynamics as a riot in the US. The purpose of this module is to automatically recalibrate the system dynamics model to different locales.

Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes (Box2)


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes

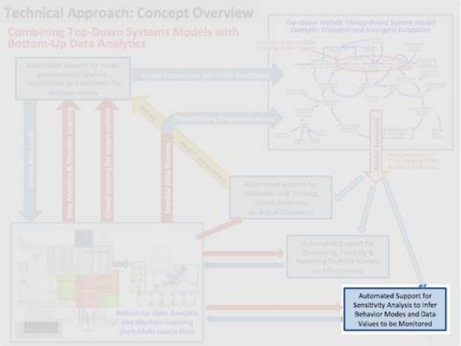
To observe how accurate the system is, the time-series forecasts generated from the system dynamics models will be compared against actual outcomes.

Automated Support for Comparing, Tracking & Balancing Multiple Models (Box 3)


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Top-Down Holistic Theory-Based System Model
Example: Dissident and Insurgent Escalation
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes
Automated Support for Comparing, Tracking & Balancing Multiple Models on Effectiveness
Automated Support for Sensitivity Analysis to Infer Behavior Modes and Data Values to be Monitored
Bottom-Up Data Analytics and Machine Learning from Multi-source Data

A system dynamics model is one method to implement a top-down module. The framework though is not limited to one implementation and one model. Multiple models in parallel can be represented. This module will determine which of the model is performing the best.

Automated Support for Sensitivity Analysis (Box 4)


Technical Approach: Concept Overview
Combining Top-Down Systems Models with Bottom-Up Data Analytics
Top-Down Holistic Theory-Based System Model
Example: Dissident and Insurgent Escalation
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes
Automated Support for Comparing, Tracking & Balancing Multiple Models on Effectiveness
Automated Support for Sensitivity Analysis to Infer Behavior Modes and Data Values to be Monitored
Bottom-Up Data Analytics and Machine Learning from Multi-source Data

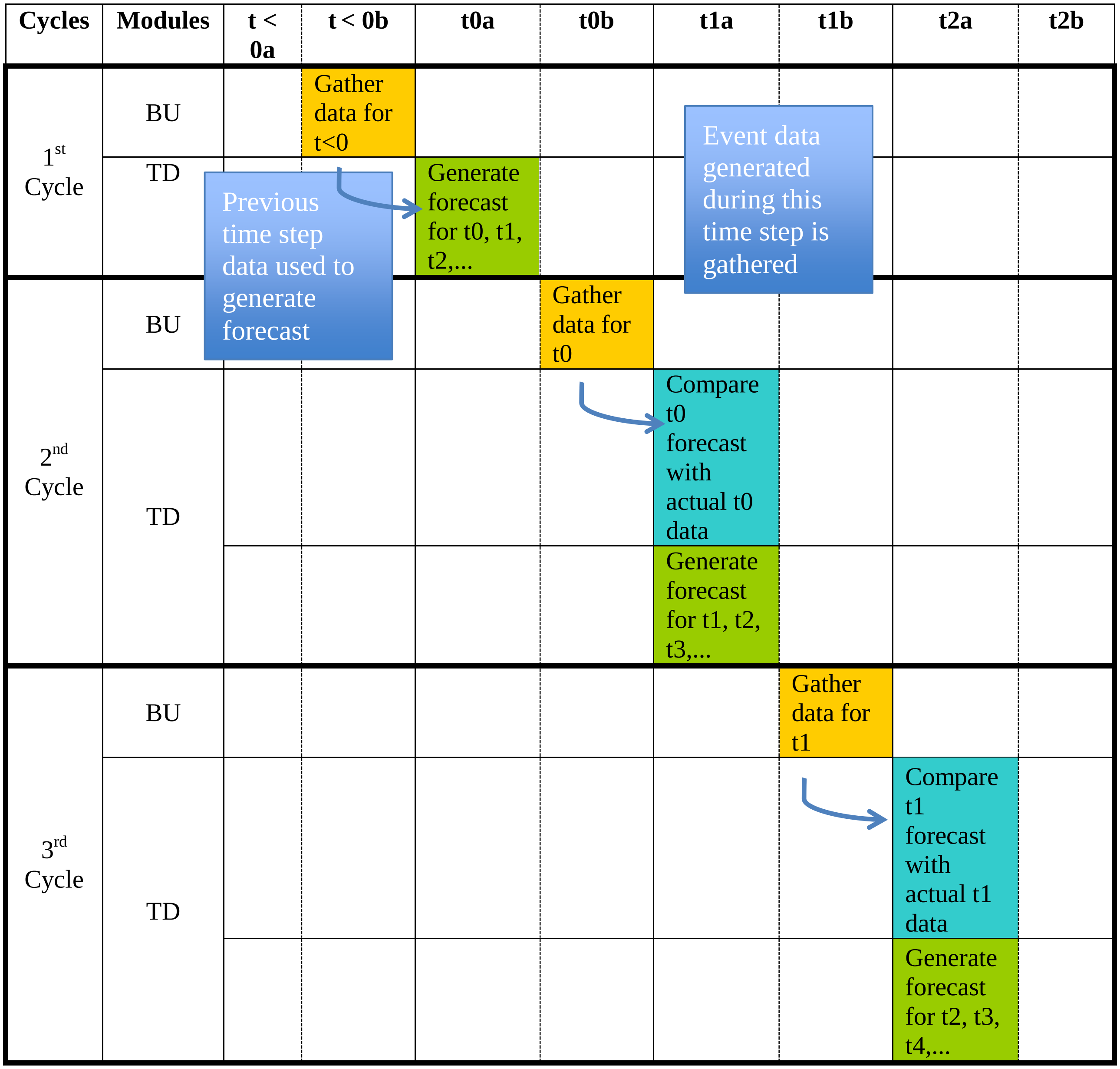
Once a system dynamics model is created it is useful to know how sensitive the output is to changes in the input. This module will automatically conduct sensitivity analysis on the system dynamics model.

3.8 Framework Data Flow

The basic data flow of the monitoring framework will be explained. The first step starts with the bottom-up module. Data will be gathered, cleaned, and processed through the data mining model. The output will be passed to the top-down module. The system dynamics model will then take the output data from the bottom-up module and start a simulation and generate a time-series forecast. This process will be repeated at a predetermined interval. At each interval the generated forecast will be compared with actual data to determine how accurate the model is behaving.

Multiple instances of this system is also possible. For example there may be four instances of a bottom-up and top-down module in parallel. The event that is being monitored will be the same, but there will be four different bottom-up and top-down models for each set. Each cycle will produce four forecasts and the model that is performing the best can be determined as the event evolves.

The figure below is a timing chart of the data flow. Each time step is split in two, “a” and “b”, to clarify the order of execution within a time step. Actions that occur in the “a” time step are execute at the beginning of the time step, while actions that occur in the “b” time step execute after “a”. For example time step t0, has t0a and t0b. Actions in t0a execute at the beginning of time step 0 and t0b executes after t0a and for the duration of the time step. Notice that the forecasts are generated by the TD module and at the start of a time step. Forecasts are generated using data from the previous time step. For example in time step t1a the TD module will generate a forecast for that time step and additional future time steps. The input data that the TD module used to generate the forecast was from the previous time step t0.



Cycles	Modules	t < 0a	t < 0b	t0a	t0b	t1a	t1b	t2a	t2b
1st Cycle	BU		Gather data for t<0						
	TD			Generate forecast for t0, t1, t2,...					
2nd Cycle	BU				Gather data for t0				
	TD					Compare t0 forecast with actual t0 data			
						Generate forecast for t1, t2, t3,...			
3rd Cycle	BU						Gather data for t1		
	TD							Compare t1 forecast with actual t1 data	
								Generate forecast for t2, t3, t4,...	

Previous time step data used to generate forecast
Event data generated during this time step is gathered

Table 2 Framework Data Flow Timing Chart

3.9 Analysis

The parts of the framework will be analyzed for functionality and automation capability. The order of analysis will be system dynamics model creation, top-down and bottom-up interface, box 2, box 3, box 1, and box 4. The connections between the modules and boxes will be discussed throughout the section. The top-down module will first analyze model creation, which will include generating a pool of potential variables, variable relationships, and bottom-up data interface.

3.9.1 System Dynamics Model Creation

The steps for creating a system dynamics model was briefly explained in section 2. Unfortunately, creating system dynamics model is not an easy task. There is considerable skill and experience required as expressed in the below statement from Richardson.

“Understanding connections between complex model structure and behavior comes, if one is skillful and/or lucky, after a prolonged series of model tests of deepening sophistication and insight.”[18]

One of the reasons that an experienced modeler is required is the source of the data used for modeling. Forrester stated that the largest pool of information used in creating a system dynamics model is the mental database that the modeler holds [19].

The mental model is the database that has traditionally been accessed first when considering possible variables. These variables may be different from one modeler to another since each will hold different mental models. However, statistical analysis and automated tools can be used to assist the designer.

One proposal to provide less variation in the pool of variables is to analyze historical data of similar events. Frequency analysis can be conducted on the words of the news articles. The resulting histogram will provide a list of words that were used in the news articles. The words with the highest counts can be the starting point for variable selection. Google and individual news websites can be a source of the news articles. Another database that can be referenced is GDELT where event dates can be found.

Knowing event dates will help against collecting false positive articles. One step above frequency analysis of words used in news articles to start variable selection is to use natural language processing to extract semantics. This will add additional words that were not included in news articles. The Predictive Analysis Today website listed SAS Text Analytics, IBM Text Analytics, and SAP Text Analytics as the top three choices for natural language processing [20].

This process of querying databases and analyzing news articles for variable selection can be automated. This will be a useful tool for the system dynamics modeler as it can be used anytime a model needs to be created.

3.9.2 Variable Relationships

The next step after generating a pool of variables is to identify the relationships among the variables. This again relies on the modeler's mental model and experience. Although there are statistical methods, that can be automated, to help identify relationships the use case is limited. For example Medina-Borja and Pasupathy have shown that a machine learning decision tree algorithm can be used to reveal relationships [21]. This will help the modeler create flows and connections between variables. However, their method requires survey data results in which all participants answers the same set of questions. It is possible to automate an online survey but the more difficult task is to get people to participate in a survey.

3.9.3 Top-down Bottom-up Interface

The interface between the data mining block and the system dynamic model needs to be defined. The data mining block provides values to the system dynamic model. Therefore, it is first necessary to know what the system dynamic model inputs are. Looking back at Sterman's system dynamics model process, the dynamic hypothesis step requires identifying endogenous, exogenous, and excluded variables. Endogenous variables are variables that are within the bounds of the system dynamic models. Within endogenous variable are three categories of variables. The first is boundary variables, which are convertors. These variables that only provide an output are typically placed at the outskirts of a model since they do not take an input. The second is internal variables. These are variables that are conceptual variables that are difficult to measure such as quality of work. Lastly are output variables. These are the stocks in the model. Stocks are accumulations within a model, but it does not necessarily mean that stocks are measureable in the real world. For this framework stocks that are measureable in the real world is required. Measureable output allows for comparison between generated data and actual data.

The boundary variables are the variables that will interface with the bottom-up module. Certain criteria need to be met for these variables. Data that is required for the boundary variables must be obtainable from the bottom-up module. Either the raw data is collected by the bottom-up module, cleaned, and processed through the data mining model or simply passed on. The boundary variable should be available from the bottom-up module at every time step. The range of values the boundary value expects and what the data mining model can produce must match.

3.9.4 System Dynamics Output Variables

The system dynamics model will produce a time-series output. The output variable must be a measurable variable that can be compared with actual data. To verify if the developed system dynamics model behaves properly simulations with historical data can be used. When in monitoring mode the forecast output will be compared with actual data. Therefore it is required that the chosen output variable of the system dynamics model also be collectable from the bottom-up module. The collected data should be the actual data that will be compared with the system dynamics output. The collected data should not require prediction or categorization through a data mining model since data mining models are rarely 100% accurate.

3.9.5 Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes (Box 2)

As explained previously forecasts are generated at the beginning of a time step for the current time step and beyond. Referring to table 2 the forecast for time step t1 is generated at the beginning of the state using data from time step t0. The generated forecast is for the current time step t1, t2, and t3. How far in to the future a forecast is generated for is determined by the user of the system.

Table 2 also shows that before the current forecast is generated, a comparison of the previous forecast is conducted. At t0a the forecast for that time step is generated. At t0b the actual data for that time step is collected. At t1a, which is the start of the time step all data from t0 has been collected. It is not possible to compare the forecasted data with the actual data. Although at t0a, forecasts for t0 to t3 were generated only the t0 portion of the forecast can be compared since t1 to t3 is in the future and actual data is yet to been generated.

The straightforward value comparison between the actual data and the forecast data may not be a sufficient comparison. All data will have some degree of measurement error included. The actual data that is collected by the bottom-up module will also include some measurement error. One possibility to determine if the actual data that includes some measurement error matches with the forecast is to use Markov Chain Monte Carlo (MCMC), which uses probability theory to determine if the given forecast agrees with the data. Actual data for a number of time series along with the initial time step from the same number of forecasts maybe required before MCMC may be used. This is an area that will require further research in the future.

3.9.6 Feedback

Control theory uses feedback to help a system target a goal. The same feedback principle can be used to assist in the numerous models in the top-down module to correct the difference between the actual data and forecasts. The comparison of actual data and the forecast will provide the difference on how off target the forecast is. This difference will be fed back as an input to the top-down module.

The difference between the actual data and the forecast data is a basic form of feedback. There are a number of more advanced feedback methodologies that can be implemented. One popular feedback method used in tracking is a Kalman filter. The exact implementation is another area where future research can address.

3.9.7 Automated Support for Comparing, Tracking & Balancing Effectiveness of Multiple Models (Box 3)

The framework is able to support multiple models within the top-down module. Each of the models will produce a forecast. Each of the models forecasts will be compared with actual data.

Different types of system dynamic models can be checked as well as other types of models such as agent-based models. A Delphi model where domain experts are polled questions can also reside in the top-down module. The forecasts that they generate will not be a continuous time-series output but they will be polled the same question at every time step and their forecast can be checked against actual results. Using multiple models will allow for determining which model is best tracking with actual data. This information will be useful for future similar events.

3.9.8 Automated Support for Model Parameter Calibration, Recalibration and Validation for Multiple Locales & Situations (Box1)

When creating a system dynamics model the pattern of the output time series is initially more important than the exact values [22]. As the model gets more refined the exact values become more important and the model will need to be calibrated.

A system dynamics model is composed of a number of stocks, flows, and converters. Each node will have either a value or an equation associated with it. The value or equation will need to be calibrated for the model to be as accurate as possible. Calibration is done by identifying the proper coefficient values in the equations of the nodes in the model. As mentioned previously historical data is used to assist and verify models. The data from one locale may differ from other locales. For example transit data will differ from the Los Angeles and New York City. Travel by automobile is more popular than public transportation in Los Angeles, where as New York City will have much more use of public transportation. Riot data from the US may differ greatly compared to riot data from the UK. The cause of a riot may be due to a governmental policy, which will be different for each country. Therefore when a model is initially calibrated for one location it may not behave properly for another locale.

Calibration is conducted in the model creation phase and done in an iterative fashion. To ease the iterative nature and recalibration of different locals the process can be automated.

3.9.9 Crowd Sourcing for Expert Opinion (Bottom-Up Output)

This arrow that is an output of the bottom-up module indicates that expert opinion has been mined electronically in some manner. There are two forms gathering expert opinion. The first is the Delphi method, which is a formal method of where domain experts are queried. Another source is the wisdom of the crowds who are not formal experts but in aggregate can provide expert level opinion. Twitter is an example of how the opinion of the crowd can be gathered. This arrow should be an input to the bottom-up module. The module is gathering data on events and crowd opinion data such as Twitter is one such data source.

3.9.10 Automated Support for Sensitivity Analysis to Infer Behavior Modes and Data Values to be Monitored (Box 4)

Sensitivity analysis is conducted at the model creation phase and can be conducted once the model has been completed to understand how changes on the input can vary the output. 50 Monte Carlo runs is recommended for sensitivity analysis [23]. Sensitivity analysis is a feature that is included in most system dynamics software tools. The output produced by many software tools will show bands where 90%, 95% and 100% of the output fell within.

With a completed model it is possible to identify which variables will affect the model the most. Ford and Flynn have shown that this can be determined when conducing sensitivity analysis. For each run the covariance between the output and input is calculated [24]. The output will show the covariance of each variable for each time step in the time series output. The covariance of variables will usually change throughout the time-series. Variables that are closer to -1 or 1 are the more relevant variables.

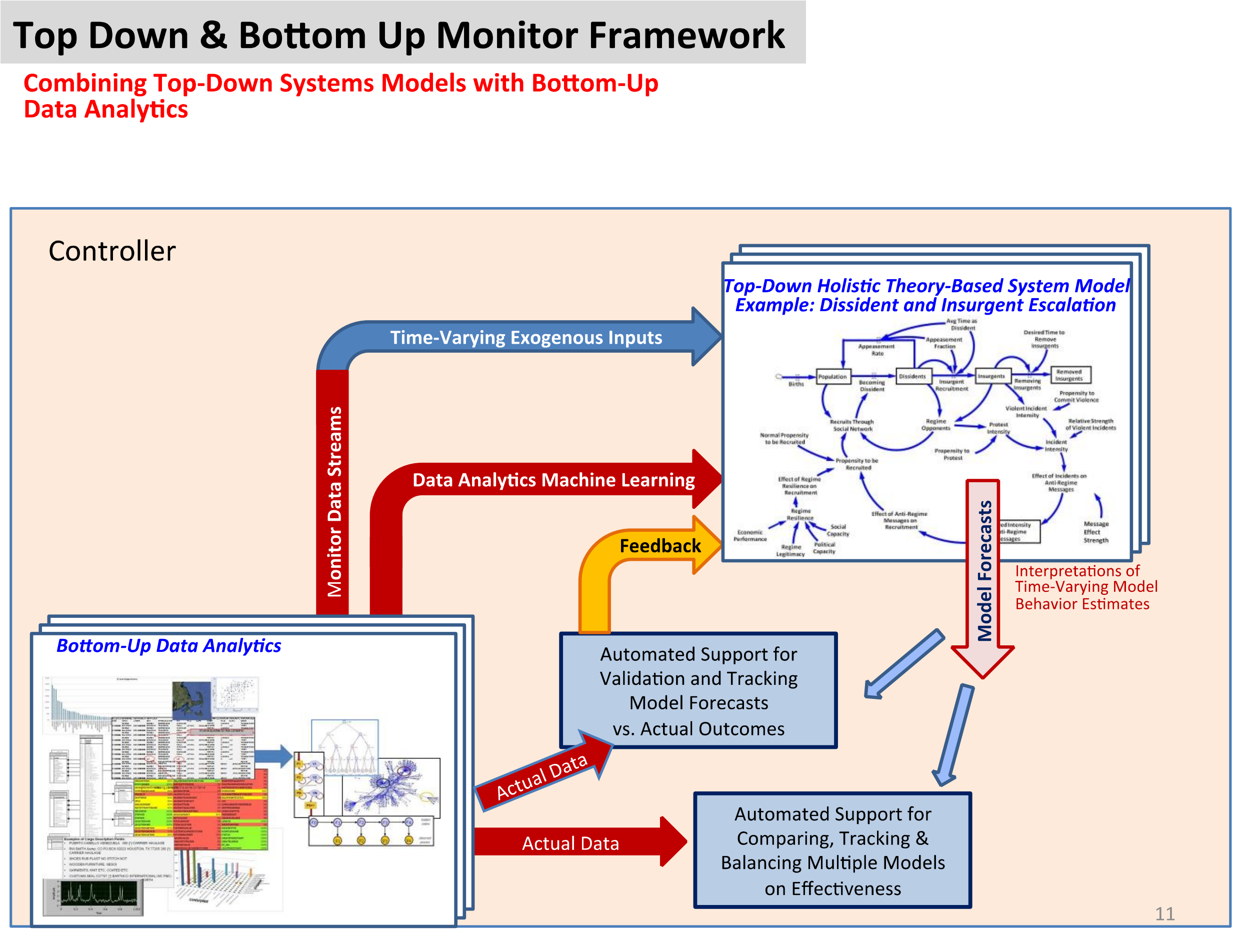
Conducting sensitivity analysis and also calculating the covariance of the variables can be automated, however this is not executed in active monitoring mode, therefore this box will be removed from the framework.

3.9.11 Controller

There is one module that is missing from the framework, a controller module. The timing diagram in table 2 show that coordination in the timing execution between the top-down module and bottom-up module is required. The controller module will instruct the top-down and bottom-up modules as well as the other boxes to execute at the correct time.

3.10 Framework Analysis Modifications

During the analysis there were two categories of activities that were identified. One category of activities is during the model creation and preparation phase. The second is activities that occur during active monitoring of an event. The framework will be modified to reflect only activities that occur during active monitoring. As a result box 1 and box 4 will be removed. This will provide a simple framework and also result in a simple software architecture. The modified framework is shown below.



Top Down & Bottom Up Monitor Framework

Combining Top-Down Systems Models with Bottom-Up Data Analytics

Controller
Top-Down Holistic Theory-Based System Model
Example: Dissident and Insurgent Escalation
Avg Time as Dissident
Appeasement Rate
Appeasement Fraction
Desired Time to Remove Insurgents
Births
Population
Becoming Dissident
Dissidents
Insurgent Recruitment
Insurgents
Removing Insurgents
Removed Insurgents
Propensity to Commit Violence
Violent Incident Intensity
Relative Strength of Violent Incidents
Recruits Through Social Network
Regime Opponents
Protest Intensity
Normal Propensity to be Recruited
Propensity to Protest
Incident Intensity
Propensity to be Recruited
Effect of Regime Resilience on Recruitment
Effect of Incidents on Anti-Regime Messages
Regime Resilience
Effect of Anti-Regime Messages on Recruitment
Economic Performance
Social Capacity
Regime Legitimacy
Political Capacity
Message Effect Strength
Time-Varying Exogenous Inputs
Monitor Data Streams
Data Analytics Machine Learning
Feedback
Model Forecasts
Interpretations of Time-Varying Model Behavior Estimates
Bottom-Up Data Analytics
Automated Support for Validation and Tracking Model Forecasts vs. Actual Outcomes
Actual Data
Actual Data
Automated Support for Comparing, Tracking & Balancing Multiple Models on Effectiveness
11

Figure 5 Modified Monitor Top-Down Bottom-Up Framework

3.11 Modifications

For this analysis, the framework has been simplified with the removal of box 1 and box 4. There are two outputs from the bottom-up module. The first is Monitored Data Stream/Time-Varying Exogenous Inputs. This is data that the bottom-up module collects from various databases each time step and is passed to the system dynamics model without any data mining model applied to it. The second is the data that has been passed through a data mining model, which includes a machine learning algorithm. Both types of data will be inputs for the top-down module. Another change is the direct feedback from what was formerly box 2. Previously the feedback was passed to box 1 and then to the system dynamics model. With the removal of box 1, feedback was connected directly to the system dynamics model input.

3.12 Riot Example

An example of how the framework will function in the case of a potential riot situation similar to the England riots will be described. The process of creating a system dynamics model, the type of data that the data mining model will gather, as well what type of machine learning models are needed will be described. Lastly the path of the data will be explained. At the end of the example a figure is provided with the details of the example within the framework diagram.

The premise of the example assumes that a police shooting resulting in a death has recently occurred and the decision to track the event has been made. The first task is to determine what the system dynamics model should forecast, which in this case is if a riot will erupt. Local authorities will be able to determine what constitutes a riot. In general it is violent actions by a crowd that can lead to destruction of property and harm to the general public. This is binary output and does not provide information on whether the situation will worsen or not. Forecast data must be compared with actual data that does not pass through a data mining model. In this case there is no actual data that indicates the trend of a situation directly. To determine the trend of the situation the location-gathering rate variable will be used as a proxy. This is the rate at which people are gathering at a particular location.

The modeler now must create a model on how a riot will occur as well as gathering patterns. There are a number of causes of riots but the focus will be on those that deal with police shootings and larger in scope government oppression. Determining the variables to be used is the next step in the model creation phase. Data mining of similar events in the past can be searched through GDELT and news articles. The found articles can be passed through a frequency counter of words to determine which are the most common. Using the words available and the experience of the modeler, relationships between variables will be created. Also at this point the modeler will determine what types of data are available that the bottom-up module can access and what algorithms can be used to transform the data in way that it can be used as an input to the system dynamics model. This will be an iterative process and after some time the modeler will produce a system dynamics model that is sufficient.

For this example the produced system dynamics model requires the number of anti-government narratives, the number of pro-government narratives, and the inferred anti-government narrative infection rate from the bottom-up module. The actual data that will be used for comparison are riot occurrence and the location-gathering rate. These are the requirements for the data mining model. The data sources that will be used are Twitter and to a lesser extent news media sites such as CNN. The data included tweets that are relevant for this example are tweet text, time of tweet, author, location, and author followers.

To determine the two types of narratives from tweets a classification algorithm will be required. In this case a natural language processing algorithm will be used. An algorithm for each of the narratives will be trained to classify the narratives to an acceptable degree. Next is anti-government narrative infection rate. The meaning of this variable is how fast the anti-government narrative is spreading. This will require a regression algorithm. Twitter users have followers. It is assumed that a tweet author has a degree of influence on their followers. Bayesian theory may be used to determine the likelihood of followers agreeing with the author’s narrative stance. This can be used develop the infection rate regression algorithm.

Lastly, actual data that will be compared with the forecast data must be gathered. The location-gathering rate can be calculated since tweets include location information. The location-gathering rate can be calculated by identifying the number of tweets that are generated at a location by unique authors over a period of time. Determining if a riot occurred or not can also be determined through tweet content and be confirmed by news media sites.

The figure below shows the details of the example within the framework diagram.
The data mining model will gather tweets from Twitter. Tweets will pass through the data mining model to determine if the context contain anti-government narratives or not. The same will be done for pro-government narratives. Second, the data mining regression model will determine the anti-government infection rate. Third, tweet content as well as news media will be check to identify if a riot has occurred or not. Lastly, the location-gathering rate will be calculated. The results of the last two will be saved while the totals for each classification and anti-government infection rate will be passed to the system dynamics model. The system

dynamics model will then execute a simulation and generate a forecast of the current state and beyond. This encompasses one time step. The forecast that was generated in this time step can be compared against the actual data collected in the next time step. The feedback will be an input to the system dynamics model. This cycle with then repeat until stopped.

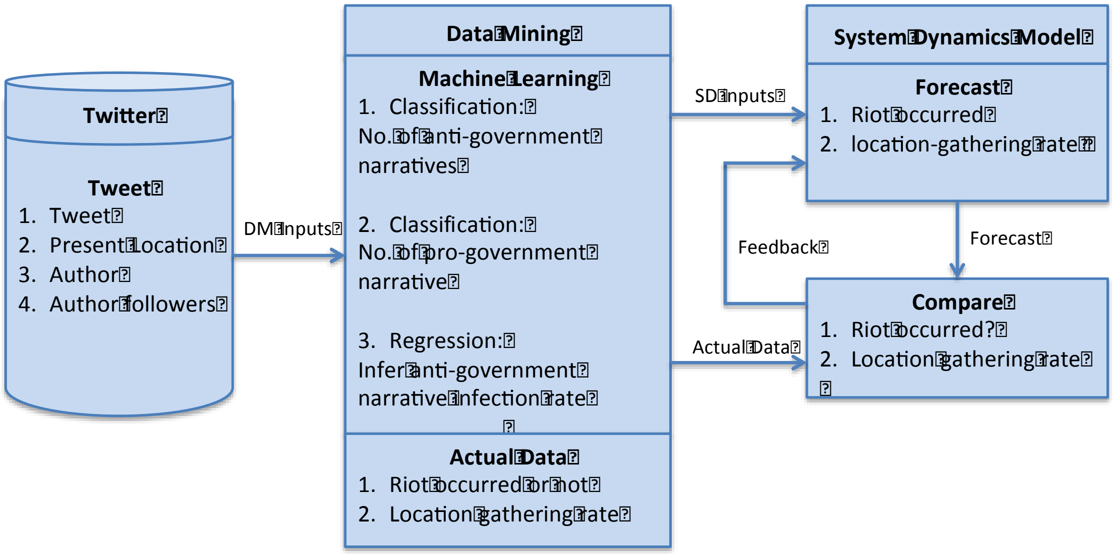


Figure 6 Riot Example

4 Top-Down and Bottom-Up Software Tools

There are various top-down and bottom-up software tools. The list of system dynamics tools which represent the top-down module were gathered from the system dynamics society website (http://tools.systemdynamics.org/). For data mining tools, which represent the bottom-up module, the list was gathered from a number of websites. The tools that were the most common among the websites were chosen. The websites used were Wikipedia (http://en.wikipedia.org/wiki/Data_mining), KDnuggets (http://www.kdnuggets.com/software/suites.html), and Predictive Analytics Today (http://www.predictiveanalyticstoday.com/top-15-free-data-mining-software/).

The purpose of analyzing the software tools is to identify which tools can be leveraged for the creation of the software architecture. Commercial and open source software tools will be reviewed for both system dynamics and data mining.

As discussed in the previous section a controller will be required to coordinate the timing of events. If available software can be implemented in the framework then the most important feature will be if the software tool can be controlled by an external application. There are two forms of external control. One is where the application used to create either system dynamics models or data mining model is directly controllable for execution through an external application. The second is where the software tool that created the model can export the model into a form that can be integrated with another program.

For commercial system dynamics tools the following attributes will be checked: cross platform support, units checking, sensitivity analysis, and optimization. Ideally the software that will be implemented will be able to run on a number of operating systems. This will allow flexibility if a leveraged tool become unusable due to rights changes or is no longer supported. The three most appealing operating systems are Windows, Linux and OSX. Units checking refers to equations that are defined in each node of a system dynamics model. The left and right side of an equation should have matching units. The system dynamics tool should have a feature to check for unit consistency. Sensitivity analysis feature should also be included. Optimization refers a feature that fits the values in the equations with the least error. The values that are optimized are either constants or coefficients. This feature is not an absolute necessity, but would be nice to have.

For open source system dynamics tools the same attributes will be checked with the addition of GUI. A graphical user interface will greatly ease the creation of a system dynamics model, especially for those who are not experienced. There are no commercial system dynamics tools without GUI.

The commercial and open source data mining attributes that will be looked for are: external control, cross platform, and GUI.

The order of analysis will be commercial system dynamics tool, open source system dynamics tools, commercial data mining tools, and open source data mining tools. At the end of each of the analysis sections a summary table will be provided of the tool criteria. Each of the criteria will be used to score each tool from 0 to 5 with 5 being the best.

4.1 System Dynamic Tools

The commercial system tools that will be analyzed are AnyLogic, Powersim Studio, Stella/iThink, and Vensim. The analysis will then continue with open source tools: Insight Maker, Pyndamics, Simantics System Dynamics, and Sphinx SD.

4.1.1 Commercial System Dynamics Tools

AnyLogic

(http://www.anylogic.com/)

AnyLogic is based in St. Petersberg, Russia. Their GUI based tool allows for three types of simulations: system dynamics, agent-based, and discrete-event. Any combination of the three can be used in a simulation. None of the other commercial tools is capable of this. The code base is written in Java, which allows it to be executed on any operating system that can execute Java. The visual appearance of the software tool is clean and modern looking. AnyLogic also has extensive animation capabilities. System dynamics models can be exported as a java applet letting it to be executed from any browser. For external control AnyLogic allows for the model to be exported as a Java application, which can then be included in another Java application for control. The Professional version is required to be able to export to a stand-alone application. Sensitivity analysis, unit checking, and optimization features are included.

Powersim Studio

(http://www.powersim.com/)

Powersim is based in Norway. This is GUI based tool and only operates on Windows. It is able to create and run system dynamics models. Sensitivity analysis, unit checking, and optimization features are included. The SDK version is required to be able to make an independent application or be integrated with another application.

Stella, iThink

(http://www.iseesystems.com/)

Stella and iThink are produced by isee system and is based in New Hampshire. The basic functionality of Stella and iThink are identical. The difference is in the documentation. Stella is focused to academics and researchers while iThink is focused on business. Unlike AnyLogic and PowerSime this tool is not as rich in the GUI display. Sensitivity analysis, unit checking, and optimization features are included. A flight simulator version of the model can be created for end-user use. Virtual knobs and sliders are added to the inputs to easily change the values. An add-on package is required to publish models for web browsers.

System dynamics models can become difficult to understand when there are multiple connections between many nodes. To help people understand a designed model Stella and iThink have a story telling feature. Starting with one node the purpose and connections are explained. Advancing to the next page will include the next node and explain its purpose. This is continued until all nodes are visible and the complete story of the model has been explained.

Lastly, models can be integrated with other applications through their isee.NET framework, which is an SDK. Although the tool is available for both Windows and OSX the external applicant integration feature is only available for Windows.

Vensim
(http://vensim.com/)

Vensim is created by Ventana Systems and is based in Massachusetts. The tool is available for Windows and OSX. Similar to Stella and iThink, the GUI display is not as appealing as AnyLogic or Powersim. All normal functionality such as unit checking, sensitivity analysis, and optimization are included. A unique feature called Causal Tracing is available. It displays a tree of variables that cause a change on a selected variable. This allows for quick understanding of a model. Simulations that take a long time can be sped up with a feature that extracts all the equations of the model and compiles them to a C program. External control of a model is possible through a DLL. The API for the DLL is well documented. The external control feature is only available for the Windows version and on the premium DSS version.

4.1.2 Commercial System Dynamics Summary

A summary of information of each application is compiled in the table below. All of the tools are capable of connecting to external applications. AnyLogic provide the easiest as the model is exported as a Java application, which can run on any operating system. Stella/iThink are also able to export models but is not as simple as AnyLogic. Any of these will be likely candidates to be used for the top-down bottom-up framework.

Tools	Platform	External Control	Version with External Control Capabilities
AnyLogic	Win, Mac, Linux	Java Export	AnyLogic Professional
Powersim Studio	Win	MS framework	Powersim SDK
Stella, iThink	Win, Mac*	isee.NET	N/A
Vensim	Win, Mac*	DLL	Vensim DSS

Table 3 Commercial System Dynamics Tools Summary

***External Control is only applicable to Windows system**

Tools	External Control	Cross Platform	Unit checking, Sensitivity Analysis, Optimization	Score
AnyLogic	Y	Y	Y	2
Powersim Studio	Y	N	Y	1
Stella, iThink	Y	N*	Y	1
Vensim	Y	N*	Y	1

Table 4 Commercial System Dynamics Tools Score

*External Control is only applicable to Windows system

4.1.3 Open Source System Dynamics Tools

Insight Maker

(http://insightmaker.com/)

This tool is unlike any of the other tools since it is web based. The site and application are managed by a nonprofit organization called Give Team. An account has to be created to be able to use the service, but the service is free of charge. The project is also open source under GPL v3 license. The focus of this tool is to make system dynamics easy and appealing to everyone. The normal box that represent a stock in a system dynamics model can be replaced with comical graphics. A storytelling feature that explains a model part by part similar to Stella and iThink is also available. Another feature is the ability to embed a model to a website or share a non-editable version of a model with a url. As for specifications, besides system dynamics models, agent-based models can also be created. Unit checking, sensitivity, and optimization features are included.

Since the project is open source it would be possible to execute the application locally. There is also a JavaScript API, which would theoretically allow for automation. Unfortunately after close inspection it would be difficult to access the API for full control of a system dynamics model.

Pyndamics

(https://code.google.com/p/pyndamics/)

Pyndamics is not a GUI system dynamics tool. It is a Python library that allows for stocks and flows to be created and simulated in Python code. There are no additional capabilities such as unit checking, sensitivity analysis or optimization. There is no traditional document that explains the usage. Instead a series of examples are provided. This project appears to be owned and maintained by a single person. There is no information any further development is being conducted.

Simantics System Dynamics

(https://www.simantics.org/)

Simantics System Dynamics is part of the large Simantics platform. It is a plugin platform that will allow simulations of other domains besides system dynamics. Simantics System Dynamics accompanies Simantics as the system dynamics modeling tool. It is a GUI tool that allows for model creation and simulation. The binary is available for Windows, but other operating systems will require installing from the source. The backend that is used for the tool is OpenModelica (see Other Modeling Tools). Units checking and sensitivity analysis features are included. Currently optimization is not supported. External control is also currently not available. A scripting feature is on the roadmap. This will allow for external control.

The community for Semantics is not large. The forum has only a few entries. However the project is managed to a near professional level. The documentation of the platform is

Sphinx SD

(http://www.sphinxes.org/)

Sphinx SD is a GUI based system dynamics tool. It allows for model creation and simulation. Sensitivity analysis is included but no other feature is available. This project does not appear to be managed actively. The website does not show any information regarding a roadmap. The forum also only has a few entries. There will be little to no support for this tool.

4.1.4 Open Source Tools Summary

Of the four open source tools Insight Maker has the best functionality. Although an API exists it is impractical for the online version. A local version would be better but the API connectivity is not clear. The next full-featured tool is Simantics, but external control is currently not available. The asterisk in the External Control column indicates that although the tools are not capable of control from an external application source code is available. The code base for each of the tools can be studied to create an external control feature. This may take considerable time and resources. All of the open source system dynamics tools either had not external control capability included or was not clear if it was possible. For these reasons all of the tools received a 0 for their score. These tools are not adequate to be used with the software architecture implementation.

Tools	External Control	Language	Latest Update	License
Insight Maker	No*	JavaScript	N/A – 02/08/2014	GPL v3
Pyndamics	N/A*	Python	N/A	MIT License
Simantics System Dynamics	Unknown*	Java	v1.7.0 – 03/15/2013	Eclipse Public License
Sphinx SD	No*	Java	V1.0 – 09/01/2013	Apache License v2.0

Table 5 Open Source System Dynamics Tool Summary

***May be possible to develop the feature**

Tools	External Control	Cross Platform	Unit checking, Sensitivity Analysis, Optimization	GUI	Score
Insight Maker	N*	Y	Y	Y	0
Pyndamics	N/A	Y	N	N	0
Simantics System Dynamics	N*	Y	Unit checking, Sensitivity analysis	Y	0
Sphinx SD	N*	Y	N	Y	0

Table 6 Open Source System Dynamics Tools Score

*** May be possible to develop the feature**

4.1.5 XMILE System Dynamics Standard

When a system dynamics model is created it can either be executed in the application that it was created in or possibly exported as either a play only model. The ideal will be able to create an application agnostic format where models created in one application can be modified in another application. XMILE is such a standard. It is an xml-based standard that describes a system dynamics model [25]. When a model is exported into this format it can be imported to any system dynamics tool to be further modified or executed. This allows ease in sharing models across different software vendors. The standard is currently at 1.0, however only Stella and iThink are implementing the ability.

4.1.6 Other Modeling Tools

Modeling and simulations are used in a wide number of fields besides system dynamics. A related field is dynamic systems. These are systems that are also based on differential equations. Dynamic systems are often mechanical systems such as a pendulum. There are a number of software simulation tools that target the dynamic systems field but since the underlying principles are similar some are able to simulate system dynamics models.

The list of tools is shown below. The details will not be explained here. The important finding is that none of the tools have an API that allows for external application control.
There are two applications to note in particular. One is TRUE. Although it is commercial software it is free of charge. A particular strength of TRUE is its 2D and 3D animation capabilities. The second is OpenModelica. Modelica is a popular non-proprietary language used to model complex physical systems. OpenModelica is the open source application that allows for creating and simulating Modelica models. The backend of Simantics System Dynamics is executed by OpenModelica.

Commercial

- Berkley Madonna (http://www.berkeleymadonna.com/)
- Dynaplan Smia (https://www.dynaplan.com/)
- GoldSim (http://www.goldsim.com/)
- Simile (http://www.simulistics.com/)
- TRUE (Temporal Reasoning Universal Elaboration) – Free (http://www.true-world.com/)

Open Source

- Ascend (http://ascend4.org)

- Exploratory Modeling and Analysis ()
- OpenModelica (https://openmodelica.org/)
- PySimulator (www.pysimulator.org)

4.2 Data Mining Tools

A search on the Internet for commercial data analytics tools will result in a long list. Open source tools will provide a shorter list. Although there were many open source data mining tools the consensuses on the top open source tools were consistent among various websites. The commercial tools had little consensus on which were the top tools. The tools that were most commonly listed on multiple websites were chosen.

The nature and size between the open source and commercial tools was revealed during the analysis. The open source tools in general a small and has a general use purpose. The commercial tools on the other hand are massive tools that can include add-on modules. The focus of all the commercial tools was on business intelligence and many of the tools appear to be identical in functionality.

The commercial system tools that will be analyzed are Angoss Knowledge STUDIO, IBM SPSS, KXEN Modeler & Scorer, Matlab, Oracle Data Mining, and SAS Enterprise Miner. Apache Mahout, KNIME, Orange, Sciki-learn, R, Rapid Miner, and Weka will follow for the open source data mining tools.

4.2.1 Commercial Data Mining Tool

Angoss Knowledge STUDIO

(http://www.angoss.com/)

Angoss is a predictive analytics tool that is focused for business use. The GUI-based software tool allows for inexperienced users to analyze data without writing any code. It is also able to import and export R data. An API is available for external control. Only Windows OS is supported.

IBM SPSS

(http://www-01.ibm.com/software/analytics/spss/)

SPSS is both the name of the company and software of a popular commercial statistical package. IBM purchased SPSS in 2009 [26]. The package is similar to Angoss in that it is focused on business. The package is spread across four families, Statistics family, Modeling family, Data Collection family, and Deployment family. The data collection family is tuned for online and paper based surveys. IBM SPSS is available for Windows, Linux, and Unix. The application can be controlled externally either through Python or VB.NET.

KXEN Modeler & Scorer

(http://www.kxen.com/Products/Modeler)

KXEN Modeler is just one of many products from KXEN. The family includes Explorer, Modeler, Scorer, Factory, Social Network Analysis, Recommendation, and Genius.

Explorer helps in cleaning and formatting data so that it is ready to be analyzed. Modeler helps in creating a model and generating reports. The tool is easy to use implementing an automated approach. Scorer is used to deploy and scale the model. Social Network Analysis is used to analyze social network websites. Recommendation is to create recommendations to users. Lastly Genius is a flight simulator for testing various models. KXEN Scorer has a number of API options available for external control. KXEN was acquired by SAP in 2013.

Matlab

(http://www.mathworks.com/products/matlab/)

When Matlab was created the target market was for control engineers. Matlab gained popularity among other fields due to its computational ability. Unlike the previously mentioned software packages Matlab is not targeted to business but to engineers. The user interface, which is console based reflects this. Since its initial release an addon tooled called Simulink has been added that does allow for graphical programing. There are no automated methods to help in creating a model. Matlab is cross platform and can be controlled through an external C/C++ or Fortran program.

Oracle Data Mining

(http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/index.html)

Oracle Data Mining (ODM) is a component of the Oracle Advanced Analytics Option [27]. ODM allows for data analytics to be conducted within the Oracle database. This eliminates the need to transfer data between a database and an analytics engine. ODM has a GUI that allows for visual programming. Modules are placed and chained together to create a solution. Each module has one specification function. ODM has an API that allows for external control through Java, SQL and PL/SQL. ODM is available for Windows, Solaris, and Linux operating systems.

SAS Enterprise Miner

(http://www.sas.com/en_us/software/analytics/enterprise-miner.html)

SAS is one of the most popular data analytics software vendors. The tool uses the same visual programming method as ODM. Nodes are placed in a chain to solve a desired problem. SAS Enterprise Miner is able to execute R code (R is a open source data mining tool as well as programming language that is discussed in the open source tool section below). The tool is also scalable to process a high volume of data. The fit model called the score model can be exported to a C or Java application. The desktop version of SAS is available for Windows and Unix.

4.2.2 Commercial Data Mining Tools Summary and Score

The table below shows the summary and score each tool.

Name	External Control	Cross Platform	GUI	Score
Angoss KnowledgeSTUDIO	Y - DMX	Y	Y	
IBM SPSS	Y - Java	Y	Y	
KXEN Modeler	Y - SAS, Java, C, PMML	Y	Y	
Matlab	Y – C/C++, Fortran	Y	N	
Oracle Data Mining	Y – Java, SQL, PL/SQL	N (Linux, Unix)	Y	
SAS Enterprise Miner	Y – C, Java	Y	Y	

Table 7 Commercial Data Mining Tools Summary and Score

4.2.3 Open Source Data Mining Tools

Below is a sampling of open source data mining software. With the exception of Matlab all of the commercial software reviewed targeted business intelligence. The focus for business is to process extremely large sets of data 24/7. Open source data mining in general is not focused on deployment. It is more for data exploration and data modeling. There are exceptions namely Apache Mahout that is focused on deployment. Data processing that requires GB and beyond is where Apache Mahout fits [28].

Apache Mahout

(http://mahout.apache.org/)

Apache Mahout is a Java library that contains machine learning algorithms that are scalable. Mahout focuses on machine learning algorithms for classification, clustering, and association. This is often used when there is a requirement to process a high volume of data. It is meant to sit on top of Hadoop, but can be used without Hadoop. Since this package is meant for high volume production use there is no GUI. Models created within Mahout are called recommenders. An external Java application must call a recommender. Mahout is mainly used on Linux, but can work on OSX and Windows.

KNIME

(http://www.knime.org/)

KNIME was developed at the University of Konstanz, Germany. It has a GUI interface to make the application easier to use. KNIME allows for visual programming. A category of nodes can be chained together to create a solution. Some of the categories available are: IO, database, data manipulation for cleaning data, data views for visualization, and mining for mining related algorithms. KNIME is popular with chemists since it includes a chemistry extension. The software structure of KNIME is modular based, which allows it to include other data mining applications such as Weka and R. Although KINE is written in Java and based on Eclipse there is no documentation of allowing external control over KINE.

Orange

(http://orange.biolab.si/)

Orange was developed at the Bioinformatics Laboratory at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia and is still under active development. Orange is a GUI based tool that is developed in C++ and Python. Similar to KNIME, Orange allows for visual programming. A toolbox of over 100 components is available that each performs a specific function. The components are chained together to create a solution. The components of the toolbox fall into Data, Visualization, Classify, Regression, Evaluate, Associate, and Unsupervised categories. Orange also has a scripting capability where models can be created in Python code by including the Orange libraries. Unfortunately there is no capability to create a model through the GUI and then export it to a Python script. The only option to create a model that can be controlled by another application is to use the Python scripting method. Binary files are available for Windows and OSX. Linux can be installed through compiling the source code.

Python Scikit-learn

(http://scikit-learn.org/stable/)

Scikit-learn is a machine learning library for Python. The library contains algorithms for classification, clustering, regression, dimensionality reduction, model selection, and preprocessing. This library requires an installation of Python. There is no GUI as it is a library and not an application. Python can be accessed through an external application. Scikit-learn is cross platform.

R

(http://www.r-project.org/)

R is a cross platform statistical package and a programming language in itself. R has one of the largest user base as the tool is one of the most popular statistical analytics tool. There are well over 5000 R packages registered on the CRAN site that provide utility services or apply different statistical methods [29]. R has a command line interface, although a number of GUIs can be added to R through extensions. The library of statistical algorithms along with its visualization ability has made the tool popular with statisticians. Models are created through code or scripts written in the R language. The popularity of R has prompted many commercial and open source data mining tools with the ability to execute R scripts within their tools.

RapidMiner

(http://rapidminer.com/)

RapidMiner is a company that provides software as a service. RapidMiner 6, the latest version, has a server-client architecture. The server portion is the for-fee service that RapidMiner provides. The previous RapidMiner 5 is open source and cross platform. This version is very similar to KNIME, but more popular. It has a GUI development environment and also offers the same visual programming capabilities. RapidMiner also developed in Java provides an API that allows for external control.

Weka

(http://www.cs.waikato.ac.nz/ml/weka/)

Weka was developed at the University of Waikato in New Zealand and one of the best known data analysis tools. It is a cross platform Java based GUI tool. It contains an extensive library of data mining algorithms to develop a model. The libraries have attracted other software tools such as KNIME and RapidMiner to be integrated in their software. Weka is developed in Java and allows for external control.

4.2.4 Open Source Data Mining Tools Summary and Score

The summary and score of the open source tool are shown in the tables below.

Name	External Control	Language	Last Update	License
Apache Mahout	Java	Java	v0.9 - 02/01/2014	Apache 2.0
KNIME	N	Java	v2.9.2 - 03/05/2014	GNU GPL
Orange	Python library	Python/ C++	v2.7 – 03/02/2014	GNU GPL
Scikit-learn	Python library	Python	v0.14 – 08/09/2013	BSD
R	Java, Perl, Python, Ruby	S	v3.1.0 – 04/10/2014	GNU GPL
RapidMiner	Java API	Java	v5.3 – 08/09/2013	AGPL/ Proprietary
Weka	Java	Java	v3.6.11 04/14/2014	GNU GPL

Table 8 Open Source Data Mining Tools Summary

Name	External Control	Cross Platform	GUI	Score
Apache Mahout	Y	Y	N	1
KNIME	N	Y	Y	0
Orange	Y	Y	Y	2
Scikit-learn	Y	Y	N	1
R	Y	Y	N	1
RapidMiner	Y	Y	Y	2
Weka	Y	Y	Y	2

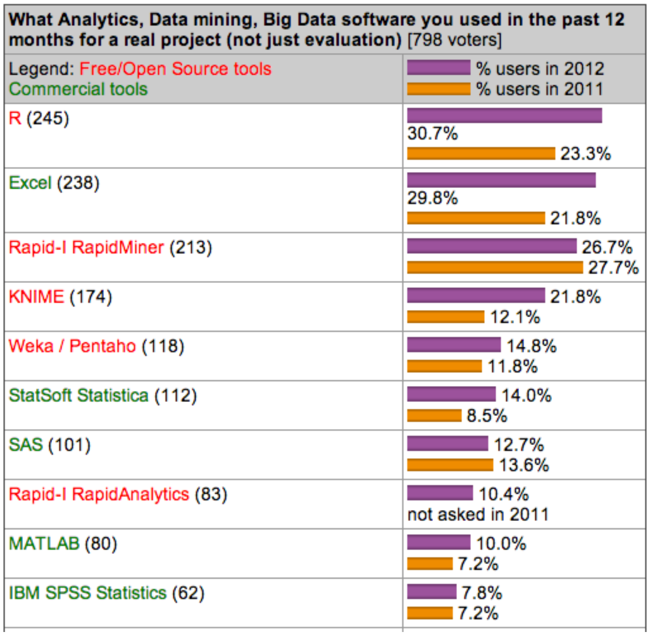
Table 9 Open Source Data Mining Tools Score

4.2.5 PMML Data Mining Standard

PMML stands for Predictive Modeling Markup Language. It is an XML standard that defines a data mining model. The Data Mining Group developed the standard with the first release in 1997. They are an independent, vendor led consortium that develops data mining standards, such as the Predictive Model Markup Language (PMML) [30]. The benefit of the standard is that a model can be developed on one application and then exported to PMML and then imported to another application for deployment. This unties the dependence on a particular application. In practice, however, the usage is not widely implemented. The DMG website has a complete list of which software tools are able to export and import PMML and at which version (http://www.dmg.org/products.html). The most recent release is 4.2, which was in February 2014. None on the list currently support 4.2 since it is relatively new. The list also shows that software tools can either export or import PMML models, but not many can do both. Also depending on the tool only certain algorithms within PMML are supported. It will be interesting to see if PMML will have more adoption in the future. Currently through the choices are limited in having a vendor agnostic platform.

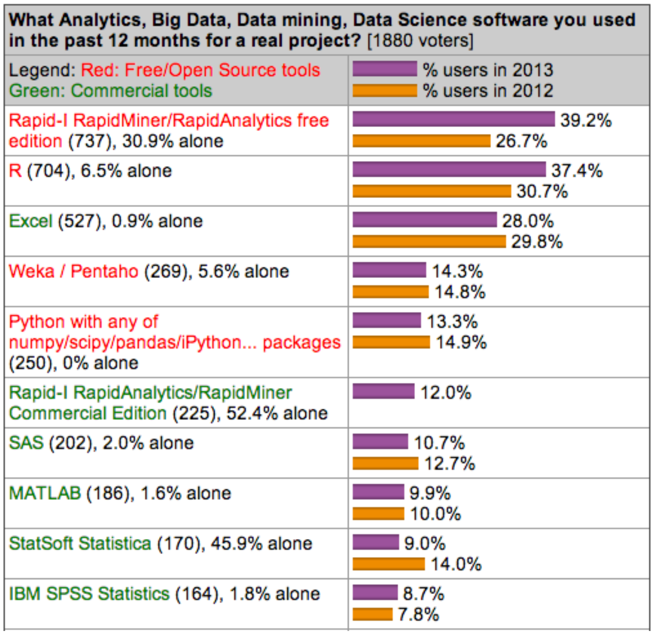
4.2.6 Data Mining Software Tool Ranking

The previous section only had a sampling of the available data mining software tools that are available. Survey data will be looked at to understand which of the tools are commonly used with data scientists. The survey data is provided by KDnuggets.com a popular website that contains a wealth of information regarding data science. The two figures below show the rankings of data mining tools for 2012 and 2013.



What Analytics, Data mining, Big Data software you used in the past 12 months for a real project (not just evaluation) [798 voters]	
Legend: Free/Open Source tools Commercial tools	🟪 % users in 2012 🟧 % users in 2011
R (245)	30.7% 23.3%
Excel (238)	29.8% 21.8%
Rapid-I RapidMiner (213)	26.7% 27.7%
KNIME (174)	21.8% 12.1%
Weka / Pentaho (118)	14.8% 11.8%
StatSoft Statistica (112)	14.0% 8.5%
SAS (101)	12.7% 13.6%
Rapid-I RapidAnalytics (83)	10.4% not asked in 2011
MATLAB (80)	10.0% 7.2%
IBM SPSS Statistics (62)	7.8% 7.2%

Figure 7 2012 Data Mining Software Tool Poll [31]



What Analytics, Big Data, Data mining, Data Science software you used in the past 12 months for a real project? [1880 voters]	
Legend: Red: Free/Open Source tools Green: Commercial tools	🟪 % users in 2013 🟧 % users in 2012
Rapid-I RapidMiner/RapidAnalytics free edition (737), 30.9% alone	39.2% 26.7%
R (704), 6.5% alone	37.4% 30.7%
Excel (527), 0.9% alone	28.0% 29.8%
Weka / Pentaho (269), 5.6% alone	14.3% 14.8%
Python with any of numpy/scipy/pandas/iPython... packages (250), 0% alone	13.3% 14.9%
Rapid-I RapidAnalytics/RapidMiner Commercial Edition (225), 52.4% alone	12.0%
SAS (202), 2.0% alone	10.7% 12.7%
MATLAB (186), 1.6% alone	9.9% 10.0%
StatSoft Statistica (170), 45.9% alone	9.0% 14.0%
IBM SPSS Statistics (164), 1.8% alone	8.7% 7.8%

Figure 8 2013 Data Mining Software Tool Poll [32]

Of particular interest between the two figures are the numbers of voters. 2013 had 1880 voters while 2012 had 798. That is over double in voters. One interpretation of this increase would be that more people are conducting data mining activities.

The two figures show that open source software is well in use for data mining. Strangely, Python is noted in the 2013 figure but not shown in 2012. The figure below, also from KDNuggtes, shows a comparison among R, Python, and other data mining tools, which are also represented each with a different color. The results consists of 562 responders however, many responders use multiple tools therefore the total users of all three categories is greater than 562. The figure shows that 26% of R users have switched to Python while 18% of Python users have switched to R. Python has retained of 79% of user while also attracting 23% of other users. What can be understood from the figure is that R has a greater user base then Python, but Python is able to retain its users at a greater percentage than R as well as attract new user, therefore Python is on a growth path.

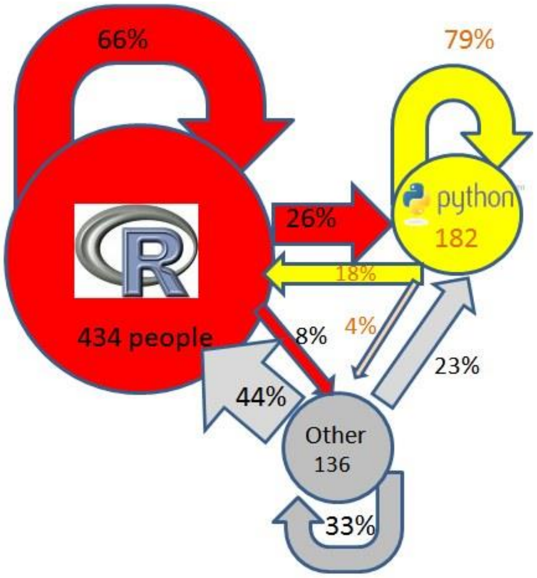


Figure 9 2013 R and Python Poll [33]

4.2.7 Candidate Tools

With the review of both system dynamics and data mining tools complete candidate tools can be listed for the monitor framework. The decision process includes the following requirements. Tools must be controllable from an external application, open source tools will be chosen over commercial tools provided that both are equal in basic capability, cross platform support, and ease of use.

System Dynamics

All commercial tools are candidates. They all have external accessibility, and have a GUI for easy model creation. None of the open source tools have the ability to be controlled through an external application. Insight Maker does have an API, but the limitations are not acceptable. API can be developed or expanded for all any of the open source tools but that will require thorough understanding of the code base for each tool.

Data Mining

All open source tools with the exception of KNIME are candidate tools. The open source tools are capable of creating a data mining model and usable for deployment. The commercials tools obviously will require a fee and are also focused on business intelligence rather than a general data mining tool. If the scale of data that will be gathered will be in the GB range then Apache Mahout is recommended since it is intended for high volume processing.

KNIME is not a candidate since there is no API that allows for external control. One advantage that KNIME has is that it is one of the few applications that can export and import a PMML model. In the future when PMML is more prevalent it may considered as a tool for creating a model and then exported to a tool that can be controlled.

5 Top-Down Bottom-Up Software Architecture

The monitor framework was analyzed and modified in section 3. Software that is used in system dynamics and data mining has been investigated. The next step is to design software architecture of the monitor framework with available software.

5.1 Previous Software Implementations

A look at what researchers have done in the past to automate system dynamic tools can provide information on what tools were used and why.

Erik Pruyt, Jan Kwakkel, Gonenc Yucel, and Caner Hamarat, have written a number papers together using system dynamics for energy related systems [34]. Their papers stated that they used a hybrid system for executing their system dynamics model. The creation and execution of system dynamic models were conducted in Vensim. Data management, analysis, and visualization were conducted with Python.

“Formerly, we used either Python or Vensim. Using both Python and Vensim together has many advantages: modeling is much easier in Vensim, but Python outperforms Vensim when it comes to making, controlling, and playing with experimental designs, analyzing and visualizing outcomes of (thousands of) simulations, etc.”[35]

The papers that Pruyt, Kwakkel, Yucel, and Hamarat wrote did not explain how they connected Python with Vensim, but the software tool that they used is available at their research group website at Delft University of Technology (TU Delft). The open source tool that they created is called Exploratory Modeling and Analysis (EMA) Workbench. This workbench as the name suggests allows for analysis of exploratory modeling. Although they used Vensim for their model in the paper the workbench is more versatile and allows for Python and Excel models [36].

Vensim documentation shows that external control can be achieved through the use of a DLL as shown in figure 5. Any application that is able to call a DLL will be able to control Vensim. The DLL documentation has a list of commands that can be executed. The available commands allows for full interaction with the model. The controls that are important for the proposed framework are: starting simulations, writing and reading data. The DLL includes these and other commands.

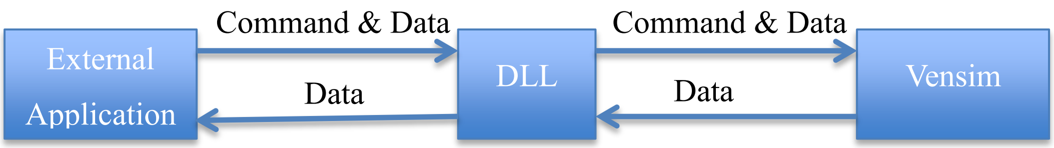


Figure 10 Vensim DLL Communication

Below is an example of the commands required to start a simulation in Vensim through a DLL. There are other commands available for more interactive experience but this example is provide to introduce the DLL command syntax.

Load and Run Full Simulation:

```
result = vensim_command(
"SPECIAL>LOADMODEL|”\Path\To\Model\SDModelToBeSimulated.vmf")
result = vensim_command("MENU>RUN")
```

The text format of the command is not difficult to understand. LOADMODEL is the command to load a model and the path and name of the model is provided as an argument. RUN is the command to execute the model. Commands fall into categories. LOADMODEL is under SPECIAL, while RUN is under MENU. Keeping this straight is the only challenging issue. The team who created the EMA Workbench made the commands easier by creating a Python wrapper. The same simulation command through the wrapper is shown below.

```
load_model(SDModel.vmf)
result = Object.run_simulation(SDModel.vmf)
result_array = Object.get_data(SDModel.vdf, Variable)
```

5.2 Software Implementation

This section will look at the software implementation of the proposed framework. This will provide information on which combination of system dynamics tools and data mining tools can be used to construct the software architecture. Construction of the software architecture has not been conducted.

5.2.1 TD/BU Connection

The EMA Workbench shows that Python can be used to control Vensim. Vensim is available in a number of versions shown below in decreasing functionality. Vensim DSS is the only version that includes external control functionality. Also the use of the DLL will only work in the Windows operating system. Therefore use of Vensim in the proposed framework must be in Windows.

- Vensim DSS
- Vensim Professional
- Vensim PLE Plus
- Vensim PLE

The monitor framework can leverage the work accomplished by the team at TU Delft and also implement a Python/Vensim tool set. To keep the language count as low as possible the proof of concept framework will implement Python for both the controller and for the data mining module.

5.2.2 Python Tools

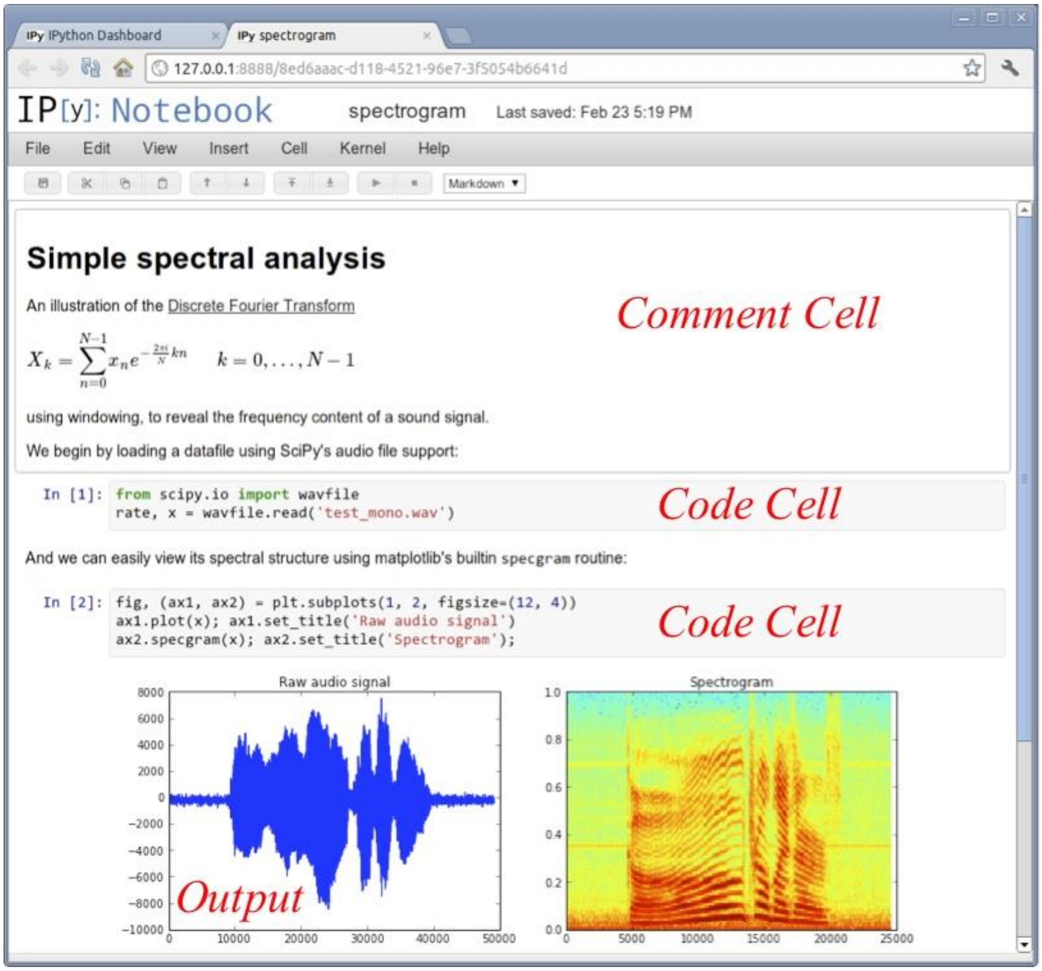
Scikit-learn is the Python library that will be used for the data mining module. The data mining model will have to be created manually through Python code. To assist in the task the SciPy package can be used. SciPy is a scientific library of Python, but it is also represents a number of additional libraries that are often used together. Below are list of these libraries.

SciPy

Fundamental library for scientific computing

iPython (iPython Notebook)

Python is a scripting language. Python programs are usually written in an editor and then executed through a command line of terminal. The iPython library allows the creation of a program be interactive within a web browser. The web browser has cells where Python code can be entered. The contents of the cell can be executed with the result displayed below the cell, which is capable of display graphs as well as text. A Python program can be written across a number of cells. Splitting the development of a program across multiple cells allows for each cell to the tested individually. Cells can also be used to provide documentation of code. The web page is referred to as a notebook since it shows the development of a program as if it was described in an actual notebook. The complete notebook can be shared easily either as a file or through a URL. Once a person receives a notebook file or link, all that is required is to open the file in a web browser and the completed program with all the cells available for execution. The below figure show an iPython notebook example. The top cell contains documentation of the program. There two cell that contain Python code and at the bottom contains the output of the program that is a spectral image.



Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Figure 11 iPython Notebook Example [37]

NumPy

A library that allows for N dimensional arrays

Matplotlib

A library to produce high quality 2D graphics

Pandas

A library that permits Python to structure data into data frames similar to R. This allows data to be analyzed within Python and not have to transfer the data to R to conduct analysis.

scikit-learn (Machine learning algorithm)

To provide more information on this library a list of algorithms that are included is listed. Clustering, Covariance, Cross Validation, Gaussian Process, Hidden Markov Model, Linear Discriminate Analysis, Naive Bayes, Nearest Neighbor, Neural Network, Support Vector Machines, and Decision Trees to name a few.

5.3 Conceptual View

A conceptual view of a proof of concept monitor system is shown below.

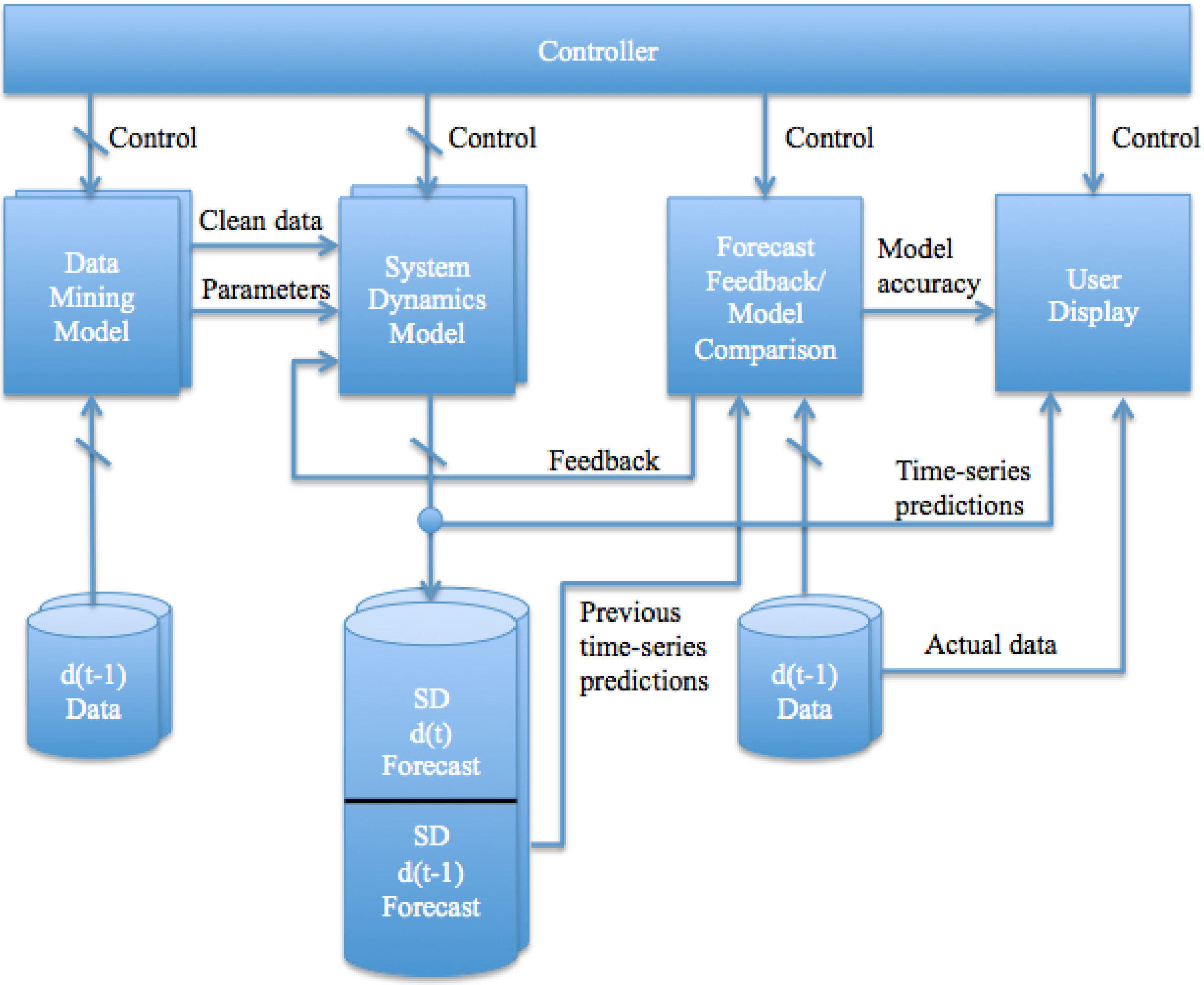


Figure 12 Conceptual Monitor Software Architecture

Note that blocks such as Data Mining Model and System Dynamics Model are two blocks each. This is to indicate that multiple bottom up and top down models can be tested in parallel. An explanation of the above blocks is provided below.

d(t-1)

This is the actual data of the previous time step. It is assumed that the data required for the data-mining block will be available.

Data Mining Model

This block contains the fitted algorithm that will be executed on the input data. The data is fetched from the data stored cleaned and the processed. The results will then be provided to the system dynamics block. Also provided to the system dynamics block is clean data. This is data that comes from the data store and is cleaned and then forwarded. This data will not be processed through the data mining fitted model.

System Dynamics Model

The system dynamics block contains the model that was developed for the target issue. The model will execute and produce a time-series output. The output is a forecast of what the future outcomes will be.

SD d(t)

This is the storage location for the forecast that the system dynamics model generated for the current time step.

SD d(t-1)

This is the storage location for the forecast of the previous time step that the system dynamics model generated.

Feedback / Model Comparison

The feedback portion compares the output of the previous time step system dynamics forecast with the previous time step actual data. The comparison method will use a feedback method such as a Kalman filter. The result will loop back as input to the system dynamics block.

The model comparison portion will compare the feedback results of multiple top down models to identify which is the most accurate.

User Display

This will display the actual data and the forecast of the current time step. Model accuracy of each of the top down models will also be provided.

5.4 Software Architecture

The below software architecture maps the conceptual design to technology that will be used. Python and Vensim are used for the architecture. The controller and data mining module are implemented in Python. Specifically the data mining model is using scikit-learn. The data storage technology is not identified as it is not core to the design. Storage can be implemented as a database or a file system.

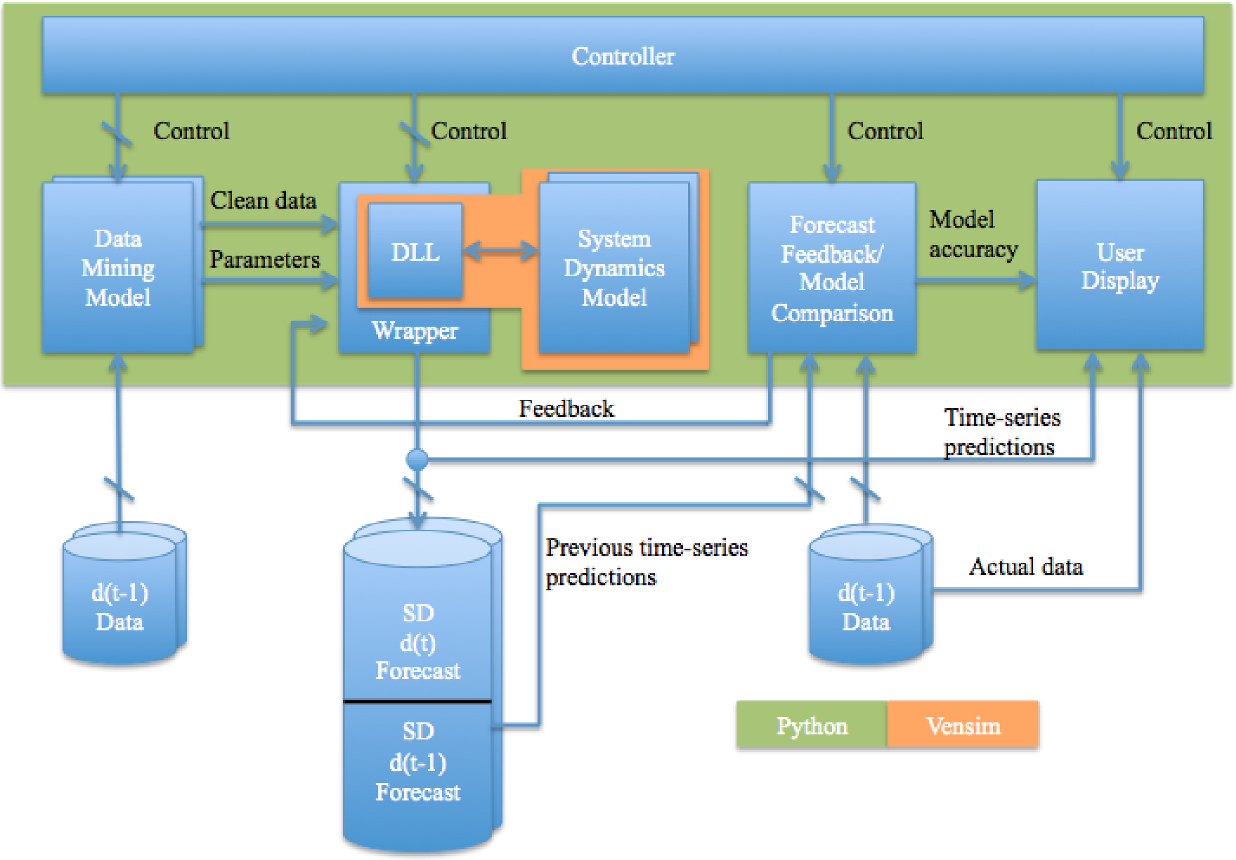


Figure 13 Python Vensim Monitor Software Architecture

The main difference between the architecture and the conceptual design is Vensim and a DLL has been included to allow for external control. The wrapper around the DLL is the Python wrapper that is used in the EMA Workbench.

5.5 Alternative Architectures

The Python and Vensim software architecture is only one of many possible combinations. Below shows a combination of architectures that can be developed for Python and Java based architectures. Any combination of data mining software tool can be used from each column with the system dynamics tools in the same column. The

Data Mining Application	**Python**	**Java**
	scikit-learn	Apache Mahout
	Orange	R
	R	Rapid Miner
		Weka
System Dynamics Applications	Vensim	AnyLogic
		Vensim

Table 10 Data Mining and System Dynamics Combinations

5.5.1 Python Implementation

The software architecture shown in section 5.4 was a Python and Vensim combination. The scikit-learn Python data mining module can be replaced with either Orange or R. The Orange implementation is similar to scikit-learn. Instead of using the scikit-learn library the Orange library can be used. The model creation method will be different but all the connections will be the same. The R implementation is also conducted by replacing the library with the RPy library.

5.5.2 Java Implementation

Java and Vensim combination will again be similar to the Python implementations. The difference will be that the controller and the data mining module will both be implemented in Java. The DLL connection to Vensim will remain. Apache Mahout, RapidMiner, and Weka have direct connection Java. R requires the JRI library for Java to access R.

The use of Java and AnyLogic will simplify the architecture. AnyLogic can export the system dynamics model as a Java application. Instead of controlling AnyLogic the model can be interacted with directly. The Java and AnyLogic software architecture is the same as the conceptual architecture.

5.5.3 Stella, iThink, and Powersim

Notice that table 9 does not contain Stella, iThink, or Powersim. These all have the ability to be controlled according to documentation. However, the detailed documentation on how to control the application is not readily available. All of them use appear to use Windows .NET framework. It is not clear if it is a matter of bridging the resulting .NET application with Java or Python. If this is the case then a Java compliant .NET framework called IKVM.NET can be used to bridge between .NET and Java. For Python, IronPython is the .NET complaint python implementation. Further research is required to verify these assumptions.

6 Conclusion

Data mining is a popular analytical method to extract knowledge from a large data set. This type of analytical method is a bottom-up methodology. It is used widely in business and academia. The popularity of this methodology has been brought on with the vast amount of data that is being generated in society on a daily basis. Another knowledge extraction methodology is a top-down method. A top-down approach starts from general principles and works down to develop models of a process. System dynamics is one form of a top-down methodology. This methodology has not garnered the same popularity as data mining, but it is considered a valuable knowledge extraction tool within the business field. Neither of the two methods is perfect. By combining the two analysis methods together it is hypothesized that greater levels of knowledge can be gained. A software tool that combines the two methodologies is required to test the hypothesis. To get to this point a framework first needed to be designed. A base framework was analyzed for functionality of the combined methodologies as well as the ability for automation. The results of the analysis produced a simpler framework. Existing commercial and open source software tools from both methodologies were also analyzed. The results of the analysis shows that commercial tools should be used for system dynamics while open source tools should be used for data mining. The modified framework was then translated into a software architecture. The system dynamic tools AnyLogic and Vensim are the easiest to integrate in the software architecture. Using Vensim the choices of data mining tools that can be implemented are Orange, R, and scikit-learn. When using AnyLogic the choices of data mining tools are Apache Mahout, R, RapidMiner, and Weka. The flexibility in choice allows for multiple combinations that can be constructed. Once built work can begin on testing the framework.

7 Reference

1. Berr and Linoff, “Data Mining Techniques: For Marketing, Sales, and Customer Support”, 1997, pg. 5
2. http://inside-bigdata.com/2013/12/18/tech-tip-power-pitfalls-clustering/
3. Shmueli, Patel, and Bruce, “Data Mining for Business Intelligence”, 2007, pg. 12
4. http://www.systemdynamics.org/DL-IntroSysDyn/origin.htm
5. http://forlearn.jrc.ec.europa.eu/guide/4_methodology/meth_systems-dynamics.htm
6. http://sloanreview.mit.edu/article/jay-forrester-shock-to-the-system/
7. Luna-Reyes and Andersen, “Collecting and analyzing qualitative data for system dynamics: methods and models”, System Dynamics Review Volume 19 Number 4 Winter 2003, pg. 275
8. Sterman, Business Dynamics Systems Thinking and Modeling for a Complex World”, 2000, pg. 89
9. Sterman, Business Dynamics Systems Thinking and Modeling for a Complex World”, 2000, pg. 95
10. Sterman, Business Dynamics Systems Thinking and Modeling for a Complex World”, 2000, pg. 103
11. http://holmes-partee.wikispaces.com/Case5-Rabbits+and+Lynx+SFD
12. Sterman, Business Dynamics Systems Thinking and Modeling for a Complex World”, 2000, pg. 551
13. http://en.wikipedia.org/wiki/2011_England_riots/
14. http://www.universetoday.com/98207/hurricane-sandy-barreling-to-eastern-seaboard-menacing-millions/
15. http://www.statisticbrain.com/twitter-statistics/
16. http://gdeltproject.org/
17. http://gdeltproject.org/
18. Richardson, “Problem with the Future of System Dynamics”, 1996, pg. 2
19. Forrester, “Policies, decisions and information sources for modeling”, 1992, 15
20. http://www.predictiveanalyticstoday.com/top-30-software-for-text-analysis-text-mining-text-analytics/

21. Mendina-Borja and Pasupathy, “Uncovering Complex Relationships in System Dynamics Modeling: Exploring the Use of CART, CHAID, and SEM”

22. Hekimoglu and Barlas, “Sensativity Analysis of System Dynamics Models by Behavior Pattern Measures”, 2010, pg. 2

23. Ford and Flynn, “Statistical screening of system dynamics models”, System Dynamics Review Volume 21 Number 4 Winter 2005

24. Ford and Flynn, “Statistical screening of system dynamics models”, System Dynamics Review Volume 21 Number 4 Winter 2005

25. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xmile

26. http://online.wsj.com/news/articles/SB124878176796786611

27. http://docs.oracle.com/cd/E16655_01/datamine.121/e17693/administer.htm#DMPRG789

28. http://www.slideshare.net/VaradMeru/introduction-to-mahout-and-machine-learning, pg7

29. http://cran.r-project.org/web/packages/

30. http://www.dmg.org/

31. http://www.kdnuggets.com/2012/05/top-analytics-data-mining-big-data-software.html#bigdata

32. http://www.kdnuggets.com/polls/2013/analytics-big-data-mining-data-science-software.html

33. http://www.kdnuggets.com/2013/12/poll-results-r-leading-python-gaining.html

34. Erik Pruyt & Jan Kwakkel & G¨onenc Yu¨cel & Caner Hamarat, “Energy Transitions towards Sustainability I: A Staged Exploration of Complexity and Deep Uncertainty”

35. Erik Pruyt & Jan Kwakkel & G¨onenc Yu¨cel & Caner Hamarat, “Energy Transitions towards Sustainability I: A Staged Exploration of Complexity and Deep Uncertainty”

36. http://simulation.tbm.tudelft.nl/ema-workbench/contents.html

37. http://ipython.org/notebook.html