Where Are You H1N1?

New Mexico Supercomputing Challenge

Final Report

April 6, 2011

Team #11
Aspen Elementary

Team Members

Kai Coblenz
Claire DeCroix
Kaan Unal

Teacher/Sponsor
Zeynep Unal

Team Mentors
David DeCroix
Cetin Unal

Executive Summary

Our project for the New Mexico Supercomputing Challenge was to continue last years H1N1 project. This year we have decided to put a detector in our model, take last years results from the survey and compare them to our school nurse, and change the spaceland to look more like Aspen school.

In our model, we set it up to look like out school and showed how the H1N1 virus spreads. We created two populations of agents, one that had bad behaviors and the other with good behaviors. The populations were split evenly through out grades K-6 and were able to move between classrooms and playgrounds where they could interact and possibly transmit the virus. We modeled the evolution of the illness when the agents collided in the simulation. The stages each agent could go through are healthy to exposed, then possibly becoming sick or becoming healthy again, and if they were exposed and became sick, they would either recover from the illness or die. If the agent recovers from the illness, they become immune and cannot become sick again, however they can become exposed and transfer the virus within the population. We tracked the number of agents in each state as the model ran, and plotted the results as a function of time.

We found that if you have good behaviors your chance of getting sick is less than if you have bad behaviors. If you have good behaviors and you are in a large group of people with bad behaviors, you are more likely to become infected because you have a higher probability of interacting with someone that is sick.

Scope of Project

In supercomputing we are studying the H1N1 influenza virus and how it spreads. We are also modeling the spread of the disease in Starlogo TNG, a computer program that implements agent-based modeling. This year we added a detector/ thermometer, and a new spaceland.

The H1N1 virus spreads from person to person, with different behaviors and habits affecting the chances of how people become infected. There are different behaviors that can affect the spreading of the disease, such as the frequency of washing of hands, coughing and sneezing into your elbow or coughing unprotected, and whether or not people share food. H1N1 is also referred to as swine flu, and its symptoms are coughing, muscle pains, weaknesses, chills, fever, sore throat, and headache. In other words, you feel pretty bad if you get it.

Reason for choosing this topic

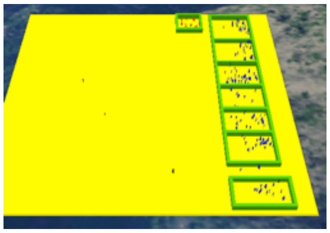
Last year team 15 did a project on the spreading of H1N1. We decided to continue this project and make improvements to it. This idea started in 2009, when the Los Alamos Middle school had a massive outbreak of H1N1, then spread through our town. The original idea was to model our school than possibly take it to the all the elementary schools, then to a county level. This however did not happen so we continued the idea of modeling our school.

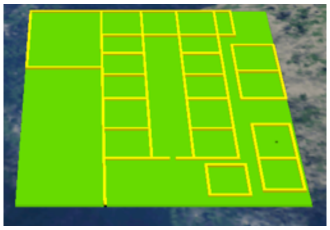
Our approach

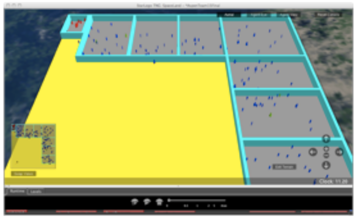
Our team went to the Kickoff Conference to learn how to use StarLogo TNG, and to hear the logic behind this computer-modeling program. After the Kickoff conference we discussed our options in further work on last years model. We decided on, trying to solve the problem of spreading H1N1. So we implemented a detector and tried to make the model more realistic by redoing the spaceland.

The StarLogo Spaceland Model

Last year we had a spaceland with seven classrooms one for each grade and a playground (bottom left). We decided that it wasn't realistic enough so we tried two classrooms for each grade, a playground, and a nurse's office (bottom center). That didn't work because Starlogo was having problems processing it all. We made a new spaceland that was similar to the first one with seven classrooms a playground and a nurse's office (bottom right).







The Detector

We decided to put a detector in our model. The detector was used to show what a difference it could make to have the sick people separated from other people so the virus could not be spread. And possibly solve the problem.

We have experimented with different detectors, (a mercury thermometer, a digital thermometer, under the arm, and in the mouth) and have come to a conclusion that a mercury thermometer is the most accurate, it is fairly cheap, but it take three minutes to get the temperature. If you want a quicker thermometer then the digital one is good. We searched for the thermometer that the doctors use and we found it was around $500, but it reads in seconds. So if you want a quick thermometer you will have to sacrifice accuracy unless you are willing to spend $500. That may not seem to be a lot, but when you add up all the classes in a school you come to $10,000 for one school. Then you have to multiply $10,000 by five to get your total for the elementary schools $50,000. So we might want to go with a cheaper detector.

We have modeled the detector with it working perfectly. This is not possible unless the schools wanted to spend about $500 on a detector. So if the detector works 90% of the time it will probably miss one or two students every day. That is still pretty good, but the sickness would still get spread through out the school. It would just go slower than if you didn't have a detector at all.

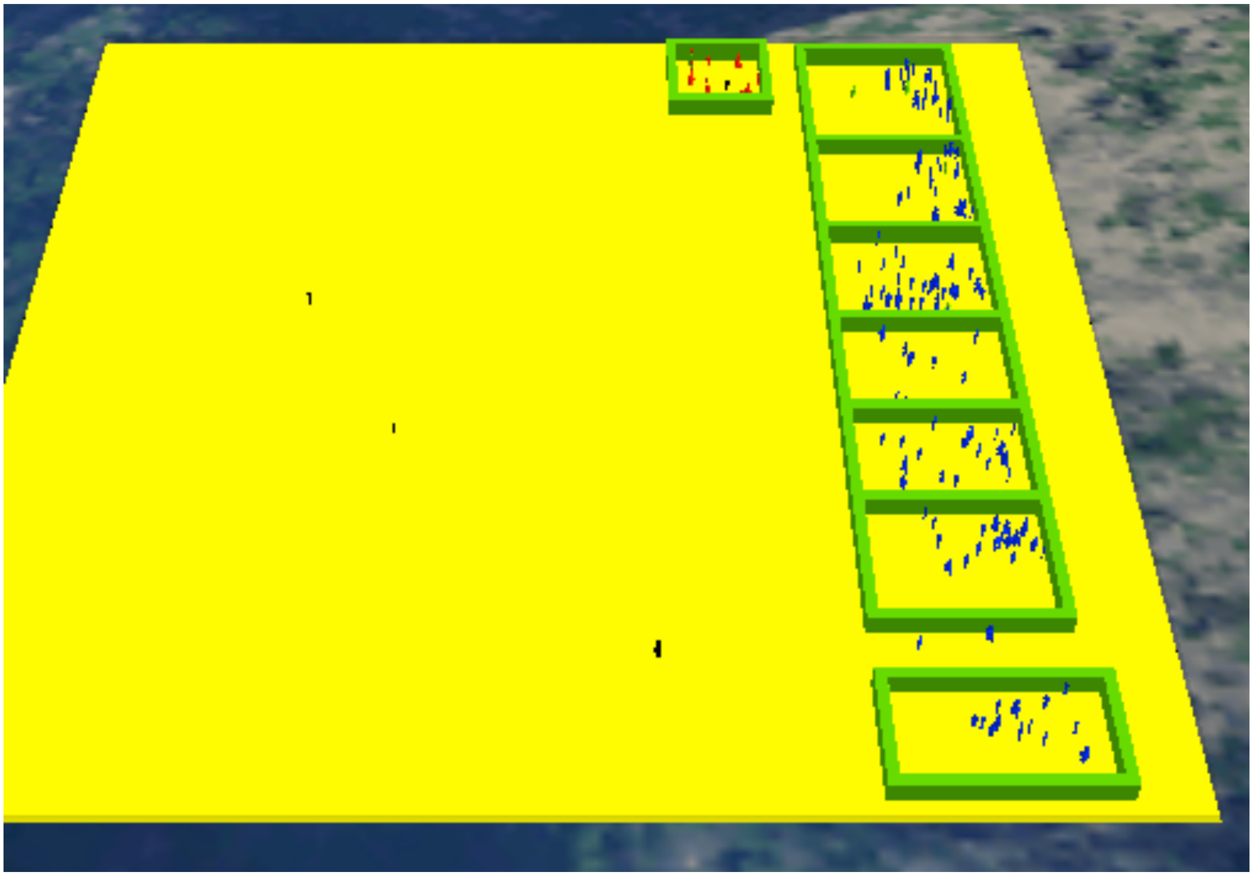


Figure 1: Agents in classrooms or nurses office.

Mobility Modeling

The model has two populations. The populations move around according to their grade (Kindergarten through Sixth grade). Each agent has a variable that tells what grade it is in. In the model the agents are either in their Classrooms/ Nurses office or at the Playground.

The two populations have good and bad agents. Each of the populations moves the same as the other. Each population is treated the same way. They are moved the same way in separate pieces of computer code. Each code contains "good" and "bad" agents-so no class has just good agents or just bad agents.

In the model we created areas. We have seven Classrooms, a Nurses office, and the Playground. We used walls to create these areas so when an agent runs into a wall it will turn 180 degrees around and will then continue moving within that area. On the spaceland all of the agents start in the Playground. After a selected amount of time the agents move to their Classrooms or the Nurses office. We used a slider that sets a variable length of time in the Classrooms or on the Playground. There are seven Classrooms – one for each grade level, and a Nurses office. Each agent goes to his or her grade level class or if they are sick they move to the Nurses office. After the class time the agents move back to the Playground.

We have set our model up so the agents go to random locations. At a time the agents instantaneously move to the Playground or the Classrooms. They are placed randomly within that area. We do this so the agents do not go on top of each other. The agents then start moving and interacting with each other.

In our model we used a clock. The clock would run without stopping. We would use the clock time for creating plots. We would use the clock time and the total amount of time on the Playground and in the Classroom in the remainder function to determine if the agents are in their Classrooms or on the Playground.

The Spreading of Influenza

The model has 2 populations. One population for agents that use good behaviors, and one population for agents, that use bad behaviors. In the model each population is it's own breed. In the spaceland we used business men to represent agents the use good behaviors and Homer Simpson to represent agents that use bad behaviors. Each population is split evenly into 7 grades, kindergarten through sixth grade. Each agent has a variable that tells what grade it is in.

The number of agents in each population can be adjusted by sliders. In our model we used a total of 160 agents. This is about one-half of the number of students in Aspen because the model only had one classroom per grade and Aspen has about two classrooms per grade. So the number of students in each classroom in the model is about the same as at Aspen.

Each agent could be in one of five different states. They are: Healthy, Exposed, Sick, Immune, and Dead. Each state has the following characteristics:

Healthy - The agent is healthy.

Exposed - The agent has been exposed to the illness.

Sick - The agent is sick.

Immune - The agent is healthy and cannot get sick again.

Dead - The agent is dead.

The model starts with about ten percent of the agents sick, and the rest healthy. We did not adjust this value in our simulations. When the simulation is running the agents can transfer the illness only by colliding with other agents.

When a sick agent collides with a healthy agent, the healthy agent becomes exposed. Then in the next time step the exposed agent either becomes sick or goes back to being healthy. The probability that the agent becomes sick, or healthy, can be adjusted by a slider.

During each time step sick agents can, either stay sick, turn immune, or die. The probability of each of these can be adjusted. This process is illustrated in Figure 3.

We stop the simulation when there are no more sick agents, because nothing there is no more illness in the populations to be transferred.

The probabilities for the simulation are adjusted by sliders. The difference between the good population and the bad population is that for the good population the chance that an exposed agent becomes sick is only 25%, while for the bad population then chance that an exposed agent becomes sick is 75%. This is an important setting for our model.

The other probabilities are the same for both the good and bad agents. The chance that a sick agent recovers and becomes an immune agent in one time step is 1%. This can be adjusted. The chance that a sick agent dies in one time step is 0.2%.

We decided to run the one simulation with three-quarters of the total number of agents (300) as good agents and one-quarter (80) as bad agents. We also ran the opposite case with 40 good agents and 220 bad agents. We will compare the results in the results section.

During the simulation we kept track of the number of healthy, sick, immune and dead agents for each population (good and bad) and each grade by using graphs.

Figure 3: A flow diagram of disease progression.

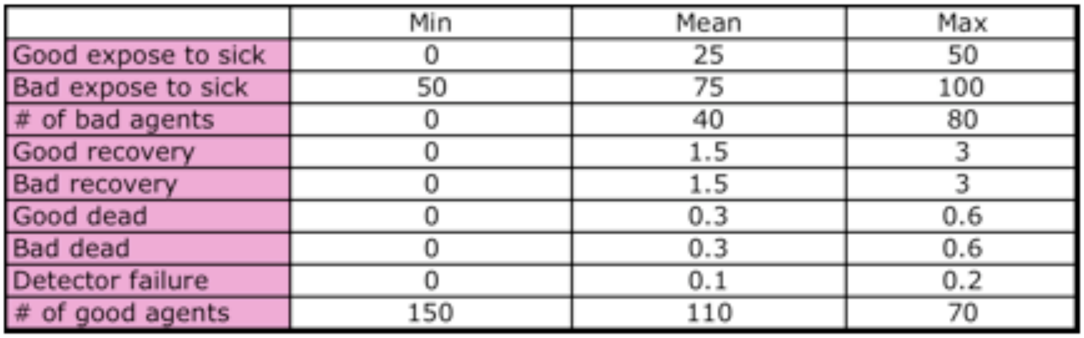
Discussion of Results

Initial Calibration, Validation and Uncertainty Quantification

In Star logo TNG all of the modeling parameters has a slider to set them up. Last year these parameters were adjusted upon expert judgment and we didn't know their minimum and maximum values. This year we would like to understand how much the simulation results would vary if we change control parameters between a minimum and maximum number. We question whether our last year's results would be reasonable or not. In order to do that we determine which potential values each parameter would take. We then assign each parameter a random number between a minimum and a maximum in order to do a Monte Carlo (Ref. 1) simulation that can give us a mean and a standard deviation for each of the calculated parameters such as the number of healthy, sick, immune and dead students there would be. The goal of this section to understand how much uncertainty would exist in the calculated parameters. This will help us to decide whether we need better calibration values for input parameters or not. The uncertainty study will also give us how good our last year's results were. Once we understand the potential range for our calculated results then we will try to calibrate our input parameters by doing sensitivity and calibration study. The sensitivity and uncertainty study presented in this section is done with the last years code in the beginning of the project. Then, we repeated the uncertainty study with the final calibrated code in the next section entitled "Final Calibration, Validation and Final Uncertainty Analysis".

We assigned a min. and max and mean to all of our 9 parameters as shown in Table 1.

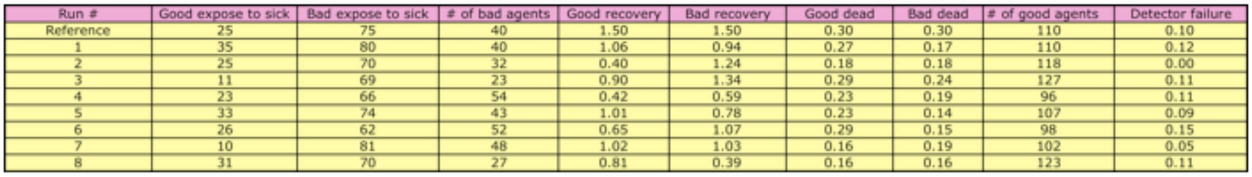
Table 1. Mean, minimum and maximum values for input model parameters.



	Min	Mean	Max
Good expose to sick	0	25	50
Bad expose to sick	50	75	100
# of bad agents	0	40	80
Good recovery	0	1.5	3
Bad recovery	0	1.5	3
Good dead	0	0.3	0.6
Bad dead	0	0.3	0.6
Detector failure	0	0.1	0.2
# of good agents	150	110	70

In order to do a Monte Carlo study we had to determine probabilistic distribution functions for each of modeling parameters. We assumed modeling parameters can change between a minimum and maximum by following a ***normal (also called Gaussian or bell-shaped distribution curve)*** distribution**.** There is no data to judge otherwise and nature usually obeys normal distribution law. Monte Carlo study requires a lot of runs. Usually the number o runs are increased until the answer does not change. Depending upon simulations this may take hundreds to millions of simulations. In order to reduce the number of runs we use a technique called Latin Hypercube Sampling (Ref. 2), this technique requires about 80 simulation runs for 9 modeling parameters each one varying between minimum and maximum. 8 of the 80 runs are shown in Table 2. Each run has different values for each of the nine modeling parameters. Latin Hypercube Sampling design generated 80 runs. We used algorithm developed by Brian Williams (Ref. 3) of LANL to determine the values of 80 runs. We are grateful to him to give us a help to avoid the large number of simulations runs that may require by Monte Carlo technique.

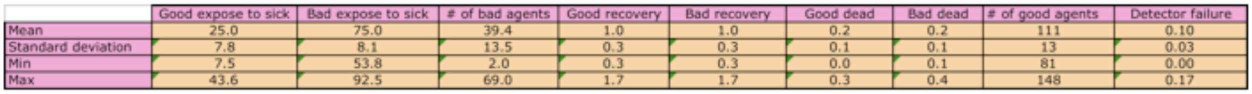
Table 2. A sample of values for nine input parameters determined by the Latin Hypercube technique.



Run #	Good expose to sick	Bad expose to sick	# of bad agents	Good recovery	Bad recovery	Good dead	Bad dead	# of good agents	Detector failure
Reference	25	75	40	1.50	1.50	0.30	0.30	110	0.10
1	35	80	40	1.06	0.94	0.27	0.17	110	0.12
2	25	70	32	0.40	1.24	0.18	0.18	118	0.00
3	11	69	23	0.90	1.34	0.29	0.24	127	0.11
4	23	66	54	0.42	0.59	0.23	0.19	96	0.11
5	33	74	43	1.01	0.78	0.23	0.14	107	0.09
6	26	62	52	0.65	1.07	0.29	0.15	98	0.15
7	10	81	48	1.02	1.03	0.16	0.19	102	0.05
8	31	70	27	0.81	0.39	0.16	0.16	123	0.11

Latin Hypercube Design gives us 80 runs. The mean and standard deviations of these 80 design values for each parameters are given in Table 3 that shows that the mean values are in agreement with the mean values we specify in Table 1.

Table 3. Average and standard deviations in statistical design.



	Good expose to sick	Bad expose to sick	# of bad agents	Good recovery	Bad recovery	Good dead	Bad dead	# of good agents	Detector failure
Mean	25.0	75.0	39.4	1.0	1.0	0.2	0.2	111	0.10
Standard deviation	7.8	8.1	13.5	0.3	0.3	0.1	0.1	13	0.03
Min	7.5	53.8	2.0	0.3	0.3	0.0	0.1	81	0.00
Max	43.6	92.5	69.0	1.7	1.7	0.3	0.4	148	0.17

We run each of these 80 cases. Each run generated a table for the number of healthy, sick, dead and immune students in both the bad and good student populations. We have created an excel sheet and calculate the mean and one-standard deviation of each output variables (the number health, sick, dead, immune students) and plotted the number of healthy, sick, immune and dead students in good and bad behavior students in Figures 1-8.

In following figures we plot last year's results with green color. This year's mean (average) simulation results are plotted with blue color. The red color shows plus minus 1-sigma standard deviation from mean values. We thought 1-sigma values would give a good value to represent uncertainty in our results. 1-sigma value represents about 75% probability that mean values would change between two red curves. The average values represent about 50% probability of our simulation results.

In Figure 1 both green and blue lines stay in 1-sigma uncertainty range. They are different at the beginning but as they progress they get significantly closer showing that last year's and this year's are both probable. These results shows that the number of healthy students in the bad population varies plus or minus ten. This shows significant uncertainty and we might have to re-calibrate this model.

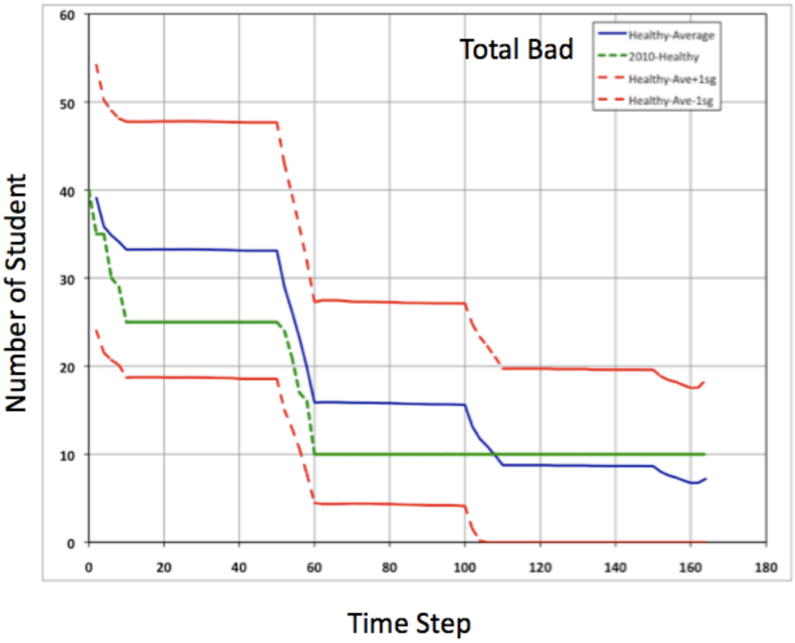


Figure 1. The number of healthy student in bad behaved student populations.

Figure 2 shows that the last year's model is questionable because their results reaches of 1-sigma range at the begining of simulation (within 20 time step). The total number of sick students varies instead of slowly going down and then it continusly drops untill it goes to zero. This is not we see in the multiple runs we did this year. The minus sigma went into the negetives so we had to reassign it to zero and both the plus.

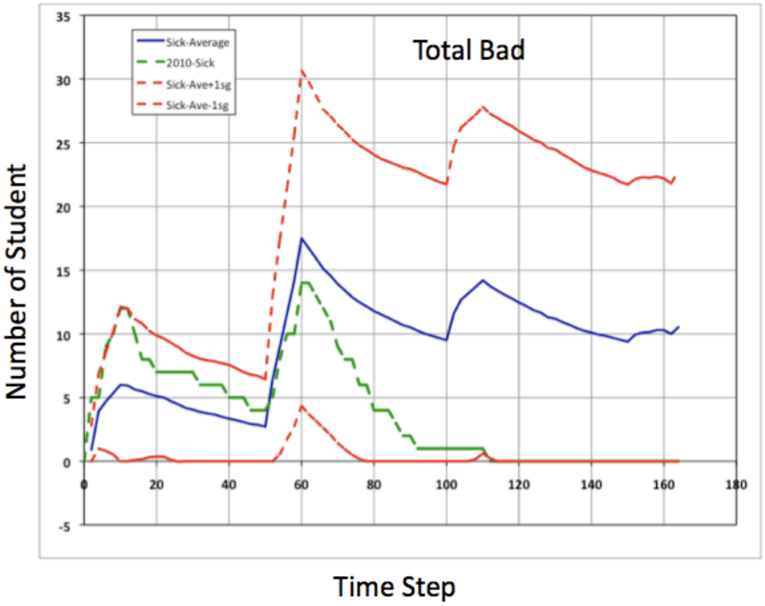


Figure 2 . The number of sick student in bad behaved student populations.

Last year’s the number of dead in the bad is very questionable (Figure 3). First, it goes out of sigma and it rises at a pace that goes up in a straight line stays at one level and keeps that flat pace for about twenty time steps and continues with constant value untill the end. This does not agree with our new results. The other difference is that old results show step wise increase in the number of dead while our new results show an smooth increase by time. This year we increased the dead rate a litlle bit more than the value they used in the last years simulations. This increase would not explain the quite different behavior we see in this figure. But defintely we can conclude that there is a bigger uncertanity in the number of dead estimations. We may need to revalidate our results using data from nurses office.

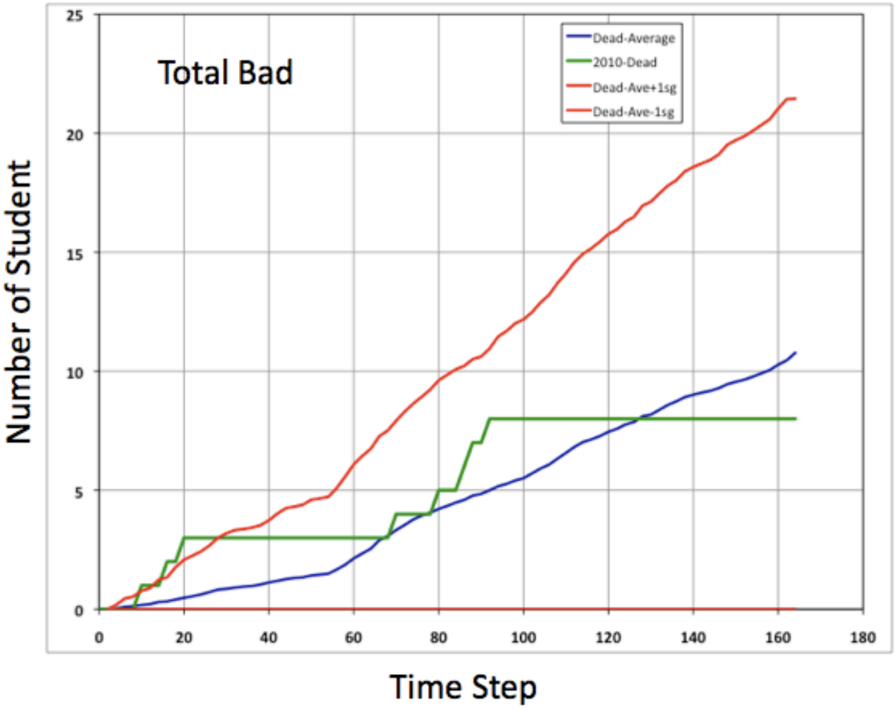


Figure 3. The number of dead student in bad behaved student populations.

Last year’s result for the number of immune students goes up straight and then levels out (Figure 4).   It goes out the plus sigma limit which makes sense because they had less sick and our results is higher because we had more sick in this year’s model so less immune.

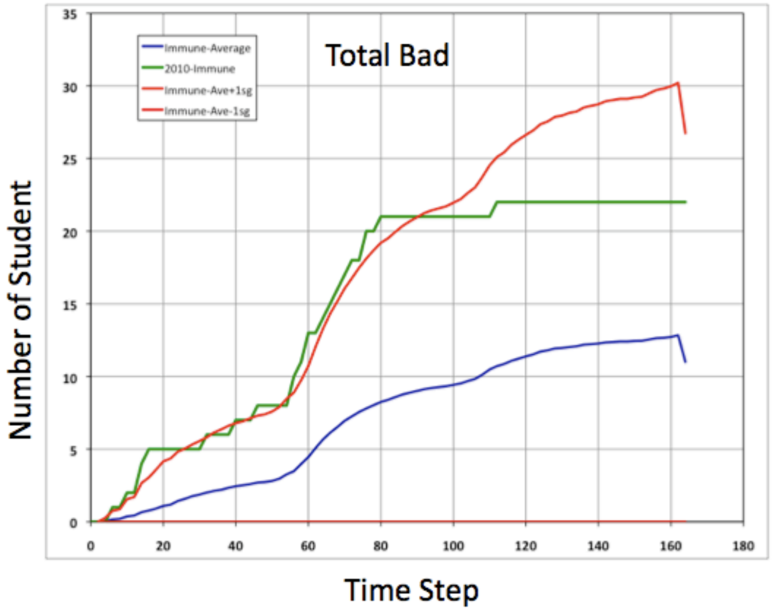


Figure 4. The number of immune students in bad behaved student population.

The number of healthy students is shown in Figure 5. Both last and this year's stay in range of sigma's and decrease in a slow manner. The last year model goes out of sigma limit by three or four students.

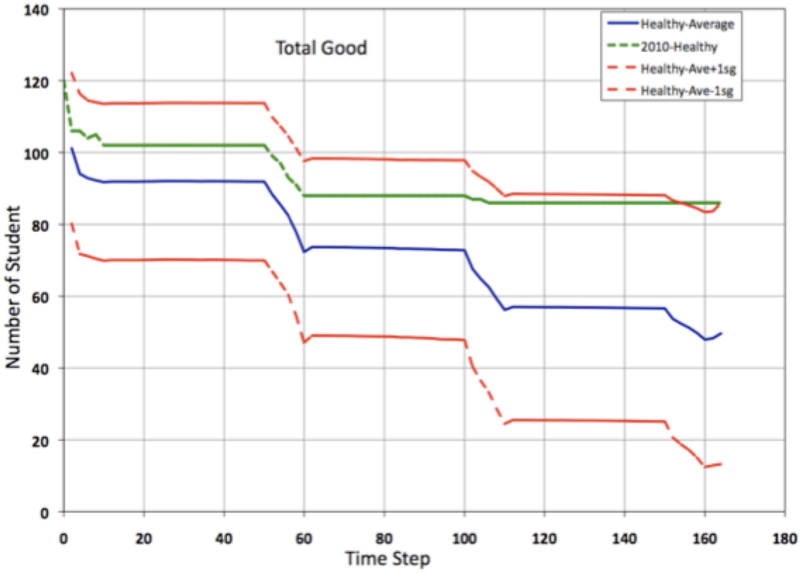


Figure 5. The number of healthy students in good behaved student population.

The number of sick students in a good behaved population is given in Figure 6. In the beginning of this test, last year's model goes out of range most of the time. However, as this year's progresses it stays in between sigma's and continues. The sick get healed and the line goes up but since the sick are mixed with the other sick they get re-infected. This year's is probable and last year's is not.

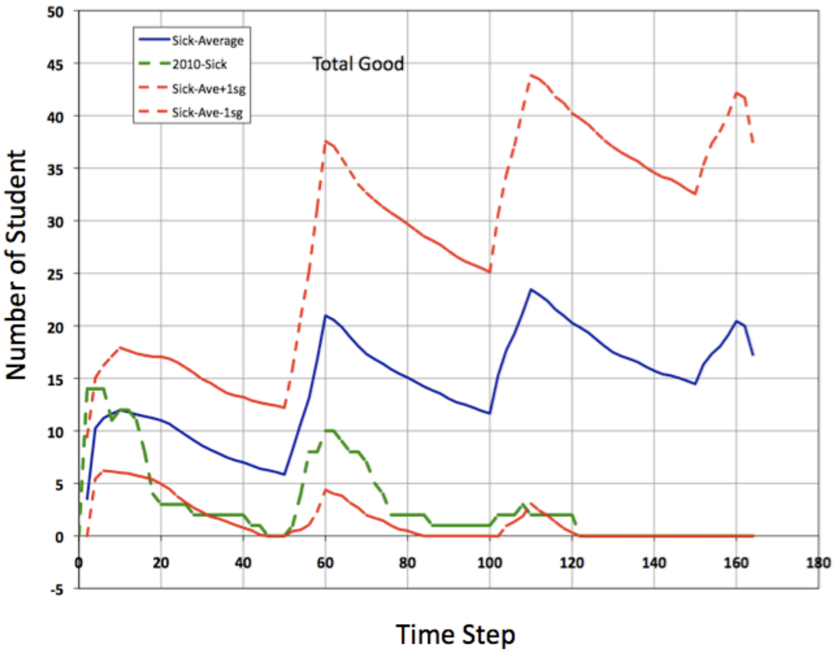


Figure 6. The number of sick students in good behaved student population.

Figure 7:Last year's model goes out of range then progresses out with a straight line and levels as our new project continues at a smooth pace and progresses like this for the rest of the model

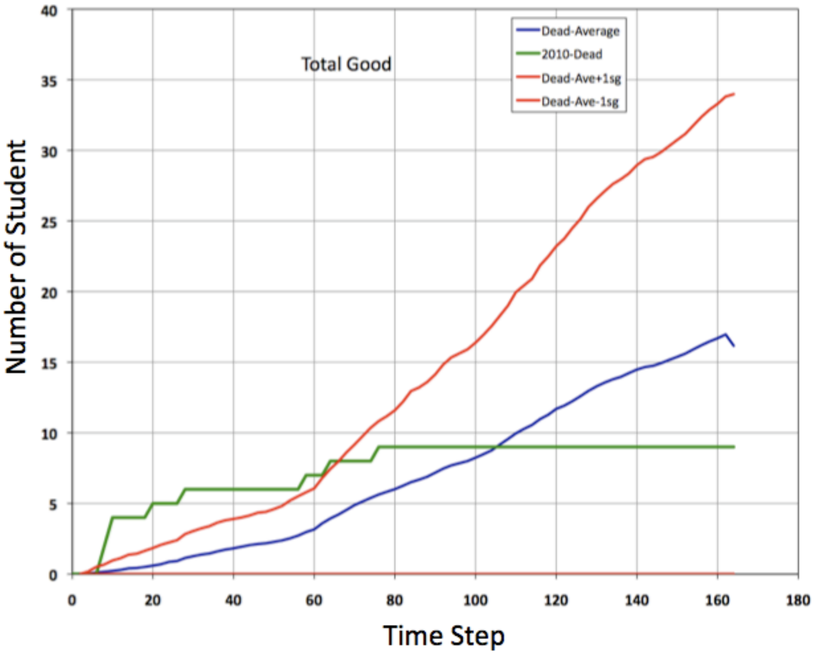


Figure 7. The number of dead students in good behaved student population.

Figure 8: This model is very similar to Figure eight last year's goes at the pace it does levels out and goes straight while ours is smooth and continues till the end.

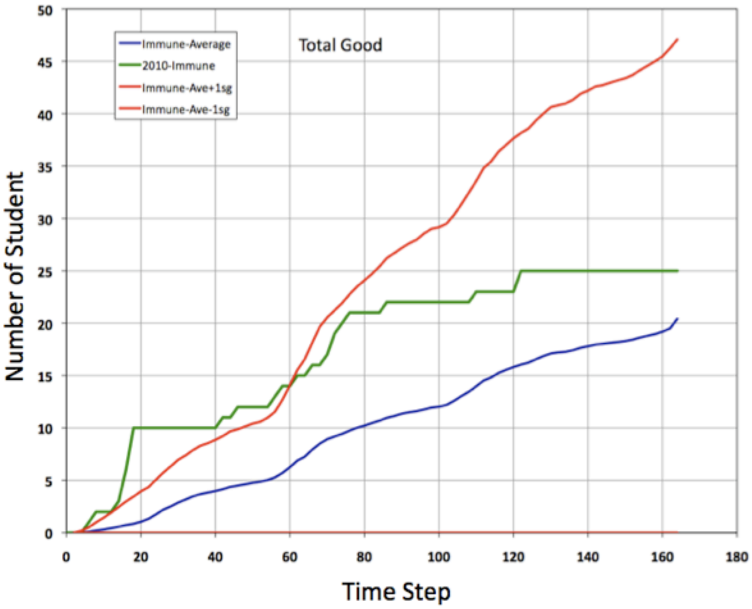


Figure 8. The number of immune students in good behaved student population.

Sensitivity Study

We did a sensitivity study because we wanted to test which parameters affected the model the most and to find out what parameters we need to calibrate. Table 3 shows the fixed parameters of the model. What we did for our sensitivity study is also shown in Table 3 (second table). Run 6 is our reference run and in the other runs each one of the parameters is doubled while the rest of the parameters stayed the same. Then, we look at results to conclude the effect of doubling an input parameter (a slider value) on the sick, healthy, immune, and dead students.

Table 3. Fixed parameters in sensitivity study and conditions in five sensitivity studies.

Good expose to sick	25
Bad expose to sick	75
# of bad agents	40
# of good agents	110

Run #	Good recovery	Bad recovery	Good dead	Bad dead
6	1	1	0.2	0.2
7	2	1	0.2	0.2
8	1	2	0.2	0.2
9	1	1	0.4	0.2
10	1	1	0.2	0.4

We summarized sensitivity results in Figure 9 that includes four sub figures. The figure in the upper left corner shows the number of healthy students in good student population as a function of time for runs 7,8, 9, 10.

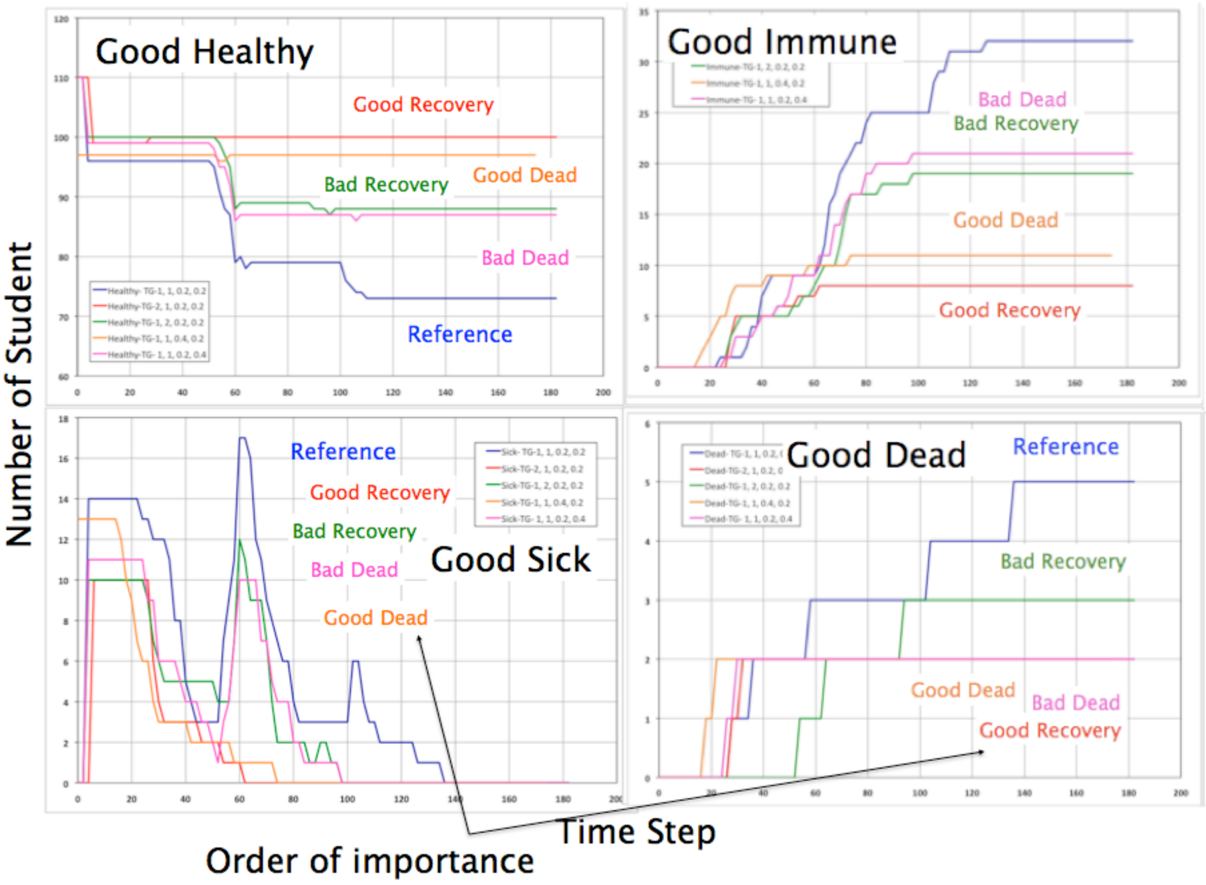


Figure 9. Sensitivity of four parameters on healthy, immune, good, and dead students in good population.

Parameter effecting number healthy students are given in an order of

- Good recovery
- Good Dead
- Bad Recovery
- Bad Dead.

In other words, if the good recovery parameter is changed twice the number healthy students increases by several factors. Second important parameter is good dead slider value and third important parameter is bad recovery slider value. The changing bad dead slider effects the number of healthy student least.

The figure in right upper corner of Figure 9 gives sensitivity results for number of immune students in good student population. The important parameters effecting number of immune students are given below in an order.

- Bad Dead
- Bad Recovery
- Good Dead
- Good Recovery

Same analysis is done for the number of sick students (bottom left corner) and number of dead students (bottom right corner). The important parameters for these outputs are given below.

Parameters effecting number of sick students are given in an order.

- Good Recovery
- Bad Recovery
- Bad Dead
- Good Dead
-

Parameters effecting number of dead students are given in an order.

- Bad Recovery
- Good Dead
- Bad Dead
- Good Recovery.
-

This sensitivity study indicates the order of importance for changes in sick, immune, dead and healthy students. The purpose of the sensitivity study is to have an idea on the ranking of importance so that in the calibration process we can cut down our trials to find out the correct slider values. From these four parameters we should be able to find a set of slider values that predicts new validation data as we will discuss in the following sections. We care the parameters affecting the number of sick students since our goal in this study to find out ways to lower the number of sick students. From the above analysis it is clear that given the fix values we used the values for good and bad recovery sliders control the number of sick students. In the final calibration study we will consider to adjust these variables first.

Final Calibration, Validation and Final Uncertainty Analysis

Validation Data

One of the difficulties to make the simulation results realistic is the lack of enough data that can be used to adjust the input parameters (slider values). In order to obtain calibration data we interviewed our school nurse and get information on the typical progression of influenza related virus infection. She indicated that one single sick student can spread the H1N1 or any influenza within a few days in the school. Historically the number of sick students can reach to 20% of the school population. After the number of sick students reaches a maximum, the sickness propagation slows down within a few days. There is usually a second way of increase in the number of sick students. However, this second peak is much smaller than the first one. She indicated in each five years one student is sent to university hospital in Albuquerque. In our initial study we interpreted this number as 1 dead per five year or 0.2 dead per year. However, we did not find any data to confirm H1N1 dead in our county. In New Mexico there was a few dead. However, our simulation is very limited and we decided to adjust our parameters to give us no dead within a week or month. If we run millions of simulation we could see a few dead. But we don't have the time to prove this. Therefore, we will lower the bad and good dead rates in our input. Finally we summarize the information is given to us by nurse below and this information is our validation data.

- Dead could be 1 per five years – 0.2 per year
- The peak sick students could reach 20 % of the school population (300)
- One sick student is enough to spread the influenza and H1N1
- In 2-3 days number of sick students peaks and comes down and picks again and gradually comes down.

Figure 10 is illustration of the validation data. Figure 10 indicates that the slider values we used in the last year simulation cannot predict the number of sick students. Therefore they needed to be recalibrated. The calibration of the sliders are discussed in the next section

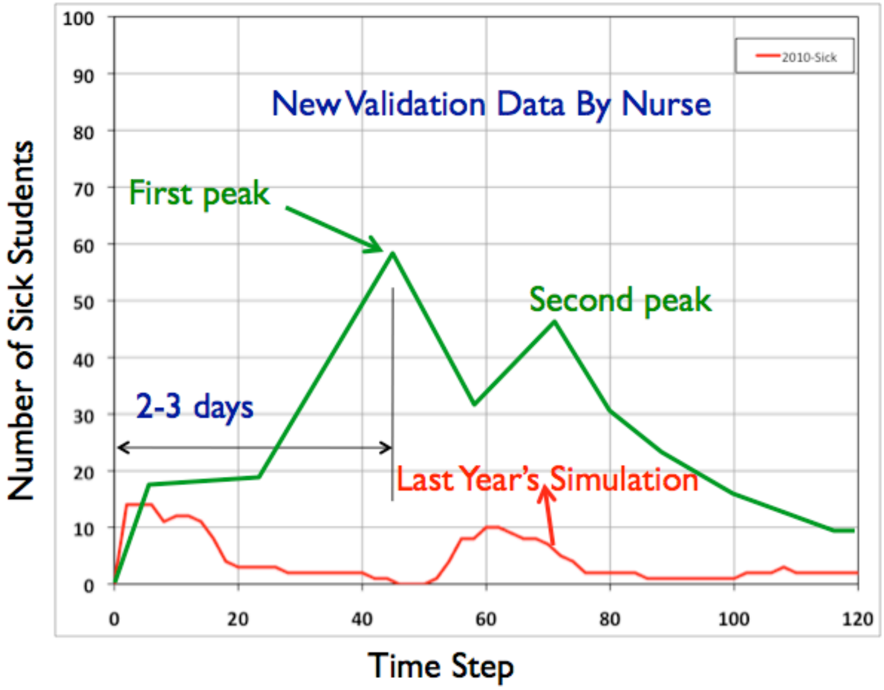


Figure 10. History of number of sick students at ASPEN Elementary School in typical influenza related sickness (Provided by school nurse and we acknowledge her contribution and appreciate her time spend with us).

Why Survey is Important

The survey is important because if you have just good agents or just bad agents then people will be mostly sick or mostly healthy. If you have just the right number of both then your model will be more accurate. Half and half is not right either because most people have good hygiene, and only some have bad hygiene. You have to find a good mix of both good and bad.

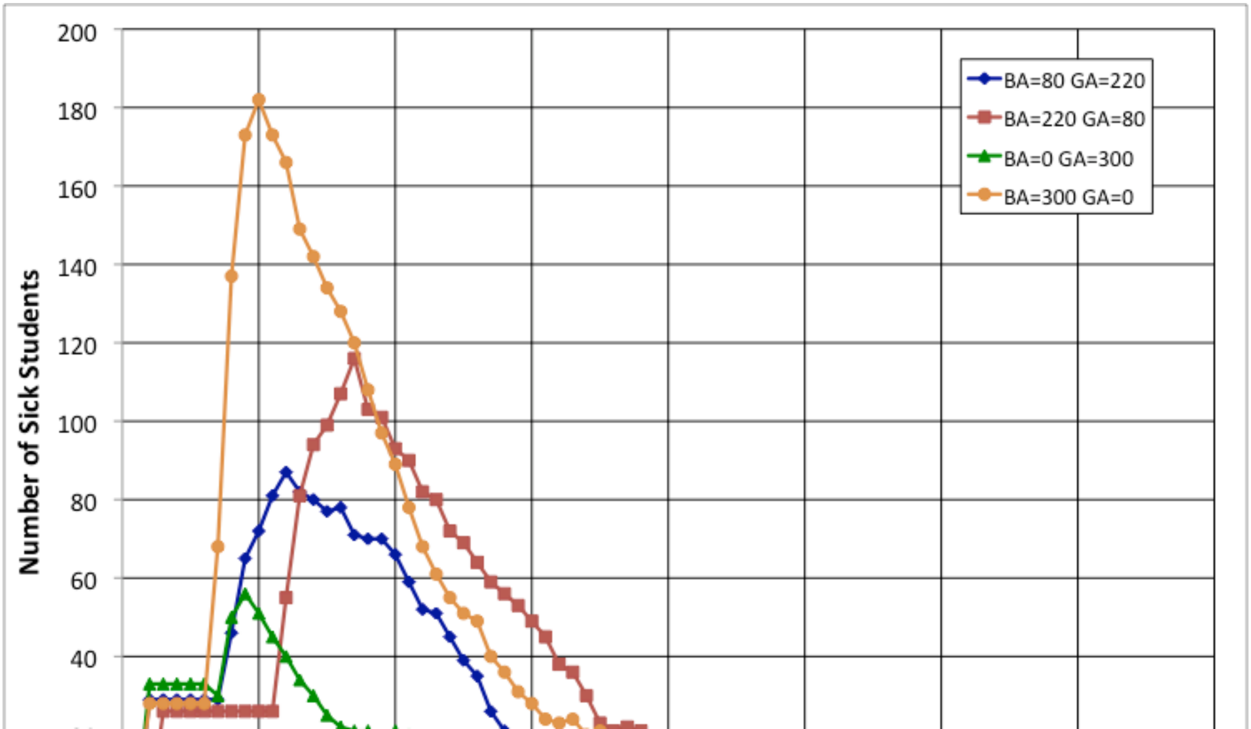


Figure 11. The number of sick students for different values of number of bad and good agents. BA=80 GA=200 is reference run. BA= bad agents, GA= Good agents.

Calibration of Slider Values

We use try and error method to find a set of input variables predicting data shown in Figure 10. After numerous tries we found following values that would give results consistent with Figure 10.

- **Good expose to sick = 5**
- **Bad expose to sick = 5**
- **Good recovery = 1.5**
- **Bad recovery = 2.0**
- **Good dead = 0.09**
- **Bad dead = 0.09**
- **Recess time =10**
- **Class time = 40**
- **Number of bad agent = 80**
- **Number of good agent =220**
- **Total number of student = 300**

Figure 3 shows the simulation logic. When sick and healthy agents collide, healthy agents become exposed. In order to decide if exposed agents become sick we draw a random number and compare it with the input value for good or bad expose to sick. If the random number is smaller than slider value then exposed agents becomes sick. There are two other random number selection processes to decide if exposed agents become immune or dead agents. These three random number drawings makes the simulation give a different answer if one repeats the simulation multiple times using the same set of slider values. Specifically, if we perform 20 runs with a given set of slider values we get 20 different answers.

In Figure 11 we illustrate this issue. Figure 1 shows six different runs with same set of input parameters given above. Although all six runs are similar in trend (the number of sick students increases to a maximum and gradually decreases by time) there are significant differences in them. Therefore we decided that any calculation we do to address a problem has to be run several times to make sure conclusions we are observing are still valid and not effected significantly by the randomness introduced by the random number generators.

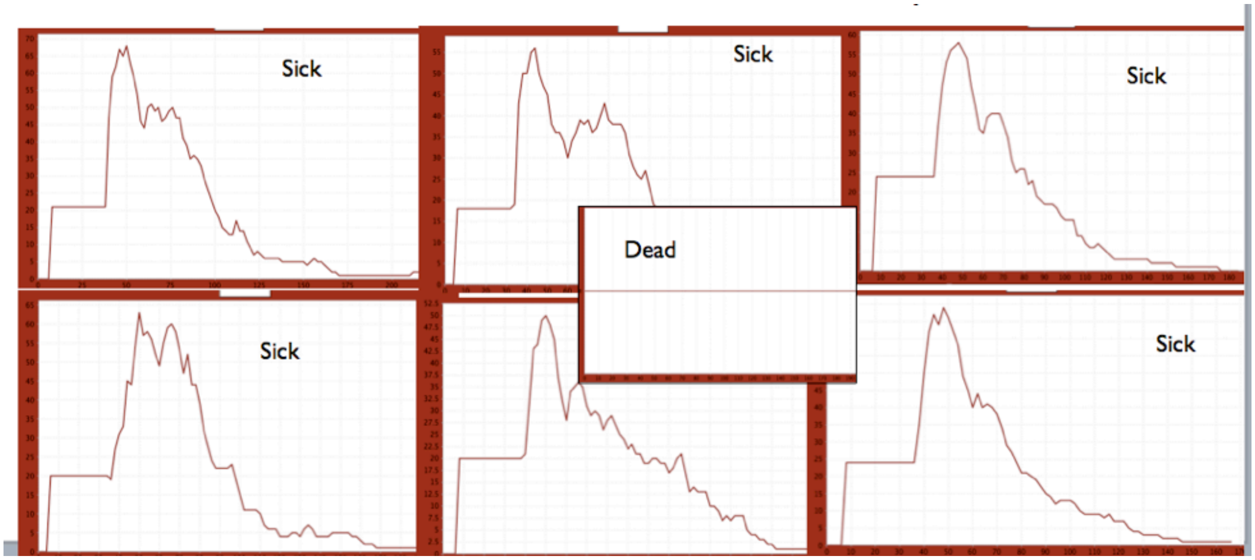


Figure 12. Repeatability of simulations with the same set of slider values. The use of random number generators results in variability in our results.

Why Survey Matters in Validation

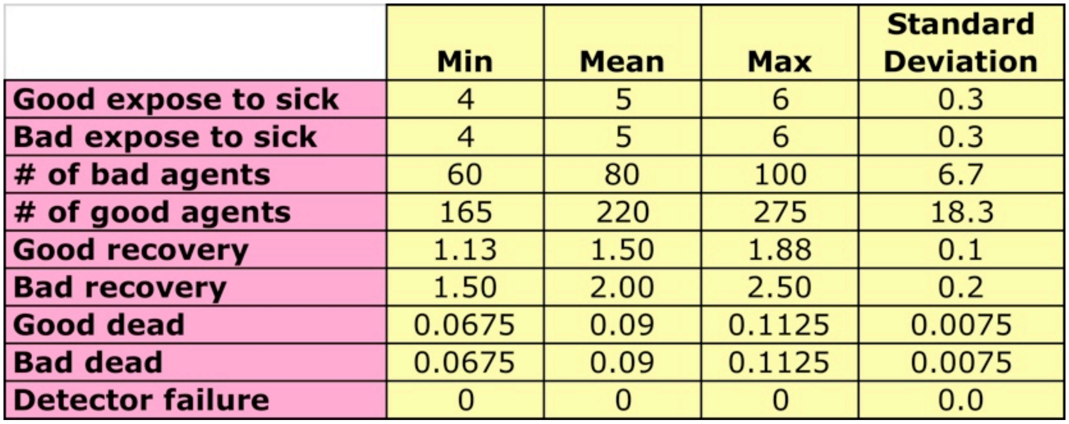
The survey is important because if you have just good agents or just bad agents then people will be mostly sick or mostly healthy. If you have just the right number of both then your model will be more accurate. Half and half is not right either because most people have good hygiene and only some have bad hygiene. You have to find a good mix of both good and bad.

Quantification of Uncertainty

As we discussed in the previous section the random number generators we used in the simulation introduces variability in our results. The use of random number generators introduces and uncertainty in our results. We will call this source of uncertainty as the uncertainty due to random number generator, UDRNG

The other source of uncertainty in our simulations comes from the selection of slider values. We adjusted these numbers to match the limited data we obtained from our nurse. However, there are multiple sliders and it is not easy to determine a unique set of slider values to reach a given answer. There are multiple solutions to get the same or similar answer. Therefore we need to assume an uncertainty range for sliders. In the first half of the year we assumed 100% error in slider values and used Latin Hypercube Sampling technique to simulate a real Monte Carlo study to determine the variability (standard deviation) in our results (number of sick students particularly). In the second half of the year we found out that the mean value of the slider values we used in the first half of the year were over predicted. They did not predict the data given us by the nurse. In Table 4 we show the new range of slider values and their standard deviations. As it is seen we narrowed down the good and bad dead rates in order to limit the dead to zero. This is consistent with data (almost no dead in last 10 years). If simulations are repeated for millions times perhaps the results will show a dead agent. Thus, the probability of dead is very low and our simulations are limited in numbers and will not show dead. We lower the good and bad exposed to sick to limit the max number of sick students to around 40-50. According to the nurse the maximum number of sick agents could be 60 that will be in 1-sigma uncertainty-range as we will discuss below.

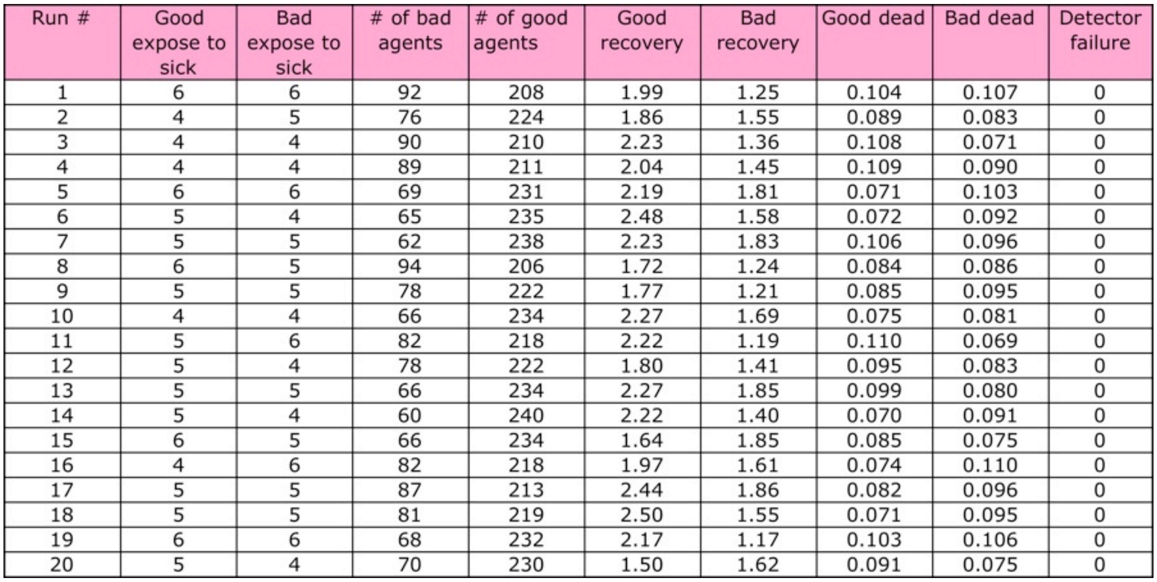
Table 4. New range of slider values showing approximately 20% variability.



	Min	Mean	Max	Standard Deviation
Good expose to sick	4	5	6	0.3
Bad expose to sick	4	5	6	0.3
# of bad agents	60	80	100	6.7
# of good agents	165	220	275	18.3
Good recovery	1.13	1.50	1.88	0.1
Bad recovery	1.50	2.00	2.50	0.2
Good dead	0.0675	0.09	0.1125	0.0075
Bad dead	0.0675	0.09	0.1125	0.0075
Detector failure	0	0	0	0.0

We were performing 80 runs according to Latin Hypercube Sampling Design. We experimented the idea of running less number of runs 10-20 and we compared standard deviation we calculated with our earlier values. We found out that our standard deviation with less number of runs did not change a lot. In fact increased and we claim that this is due to the random number generators we used in the simulations. In other words the uncertainty due to the random number generators and unknown slider values was much higher than the effect of the number of runs. Therefore we decided about 15 runs to quantify the uncertainty due to both random number generators and parametric range of slider values. Table 5 shows typical values we used in quantifying the uncertainty.

Table 5. New values used in sliders in limited Monte Carlo study.



Run #	Good expose to sick	Bad expose to sick	# of bad agents	# of good agents	Good recovery	Bad recovery	Good dead	Bad dead	Detector failure
1	6	6	92	208	1.99	1.25	0.104	0.107	0
2	4	5	76	224	1.86	1.55	0.089	0.083	0
3	4	4	90	210	2.23	1.36	0.108	0.071	0
4	4	4	89	211	2.04	1.45	0.109	0.090	0
5	6	6	69	231	2.19	1.81	0.071	0.103	0
6	5	4	65	235	2.48	1.58	0.072	0.092	0
7	5	5	62	238	2.23	1.83	0.106	0.096	0
8	6	5	94	206	1.72	1.24	0.084	0.086	0
9	5	5	78	222	1.77	1.21	0.085	0.095	0
10	4	4	66	234	2.27	1.69	0.075	0.081	0
11	5	6	82	218	2.22	1.19	0.110	0.069	0
12	5	4	78	222	1.80	1.41	0.095	0.083	0
13	5	5	66	234	2.27	1.85	0.099	0.080	0
14	5	4	60	240	2.22	1.40	0.070	0.091	0
15	6	5	66	234	1.64	1.85	0.085	0.075	0
16	4	6	82	218	1.97	1.61	0.074	0.110	0
17	5	5	87	213	2.44	1.86	0.082	0.096	0
18	5	5	81	219	2.50	1.55	0.071	0.095	0
19	6	6	68	232	2.17	1.17	0.103	0.106	0
20	5	4	70	230	1.50	1.62	0.091	0.075	0

We used the slider values given in Table 5 and repeated runs 20 times. The number of sick agents as a function of time is plotted in Figure 12. As shown in this figure the number of sick students peaks to a value around 65 in 30 steps (roughly 2 days) and gradually drop. The standard deviation in number of sick agents peaked to about 15 and dropped. We will use the green curve given in the figure as the representative standard deviation to characterize the effect of random number generations in our simulations. Note that the mean value of the maximum number of sick people is above 60. We readjusted slider values to lower this maximum to values 40-50 when we study the effect of uncertainty in slider values.

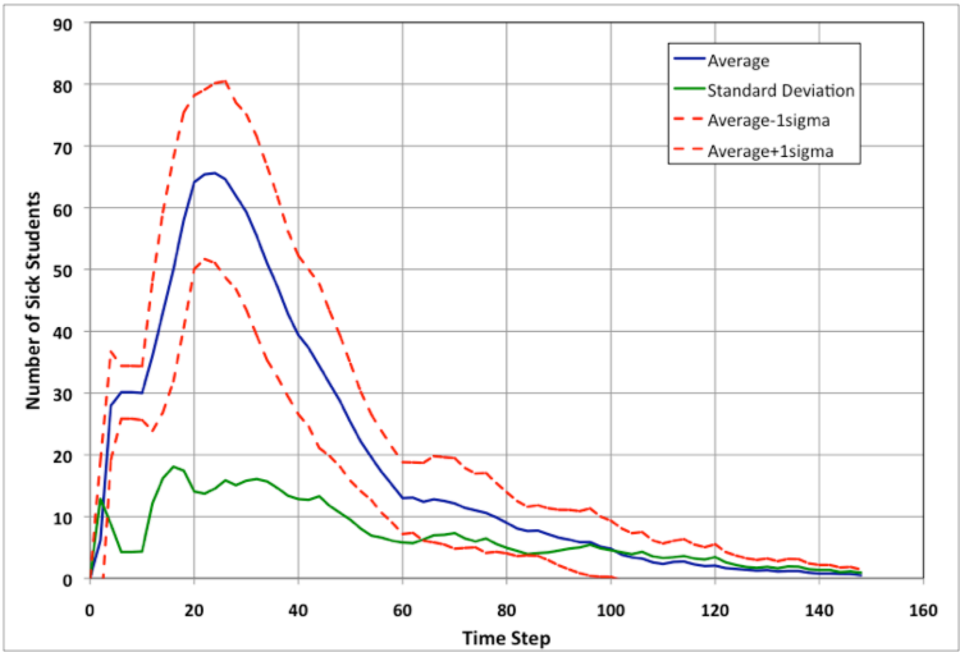


Figure 13. Uncertainty due to random number generators.

Figure 13 shows the average typical values of sick, immune, and healthy agents. These results are obtained from one of 20 runs in which we changed the slider values randomly within plus minus 20%. The number of sick students increases to value 40-50 and gradually drops. First few days drop is not significant but later agents gets immune and number of sick agents starts drop more to almost a few agents. There were no dead agents in these simulations.

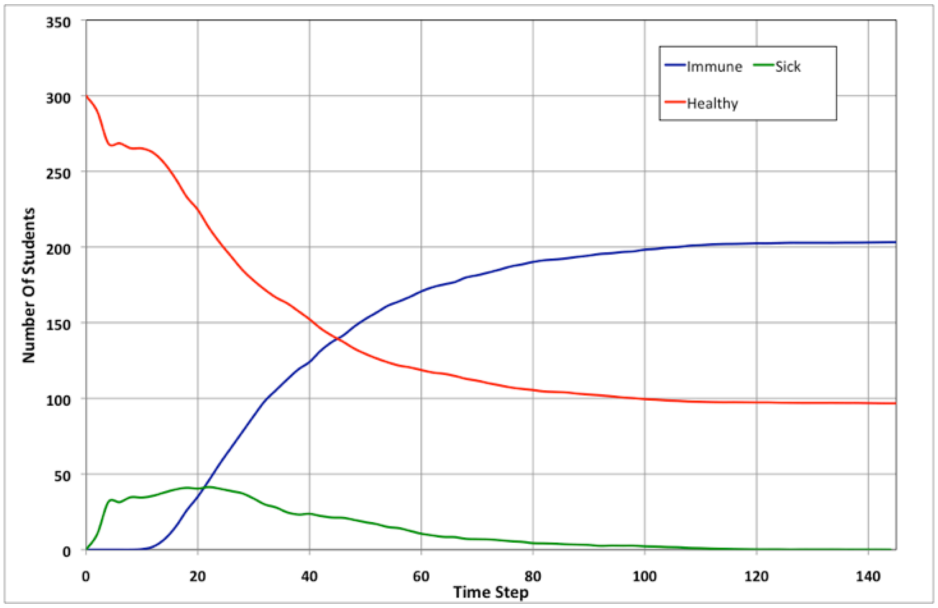


Figure 14. The typical values of number of sick, immune, and healthy agents in Monte Carlo simulations varying slider values within 20%.

Figure 14 shows the average (mean) values of number of sick agents in our simulation of20 runs. The peak of the sick agents is about 40. The number of sick agents drops and shows another mild peak and finally drops to a few values. Note that we did run our simulation 150 time-steps where the number of sick agents became almost zero. The standard deviation of these 20 runs is also shown in Figure 14 with solid green colored line. The dashed green colored line is the standard deviation due to random number generators. We assumed these two standard deviations are related with each other. For one set of slider values we could repeat runs 20 times. But this will require to run 400 runs. Instead we calculate the standard deviation of one set of 20 runs using same set of slider values. Then we used following formulate to calculate a total standard deviation representing the total uncertainty in our simulations. This formulate came form LANL scientist Brian Williams (we wish to express our gratitude for his help) and is valid to sum two dependent uncertainty contributors.

$$\sigma_{Total} = \sqrt{\sigma^2_{Uncertaint\,yInSliderValues} + \sigma^2_{RandomNumberGenerators}}$$

Where σ is the standard deviation in our simulation.

The total standard deviation calculated from this formula is shown with gold colored line. The peak standard deviation is around 25 and drops gradually as simulation progresses. We add and subtract 1 standard deviation to/from mean values to define the minimum and maximum range for the sick agents. The peak number of sick agents given by nurse (60) is within minimum and maximum range (within 1-standard deviation of simulations.

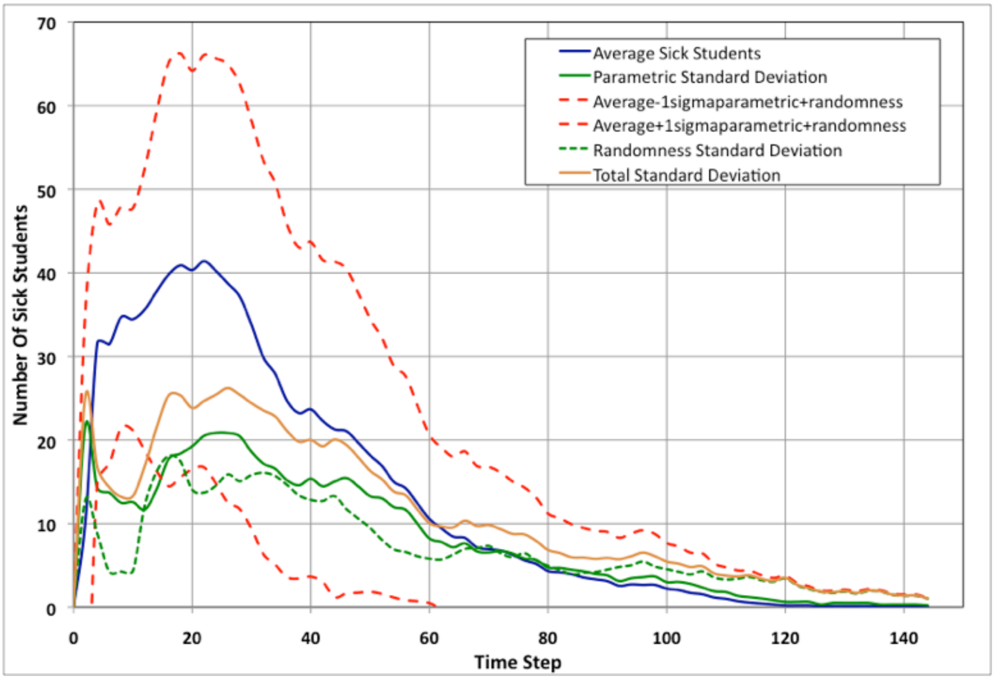


Figure 15. The average (mean) sick agents and its total uncertainty.

Effect of Detector on Spread of Influenza

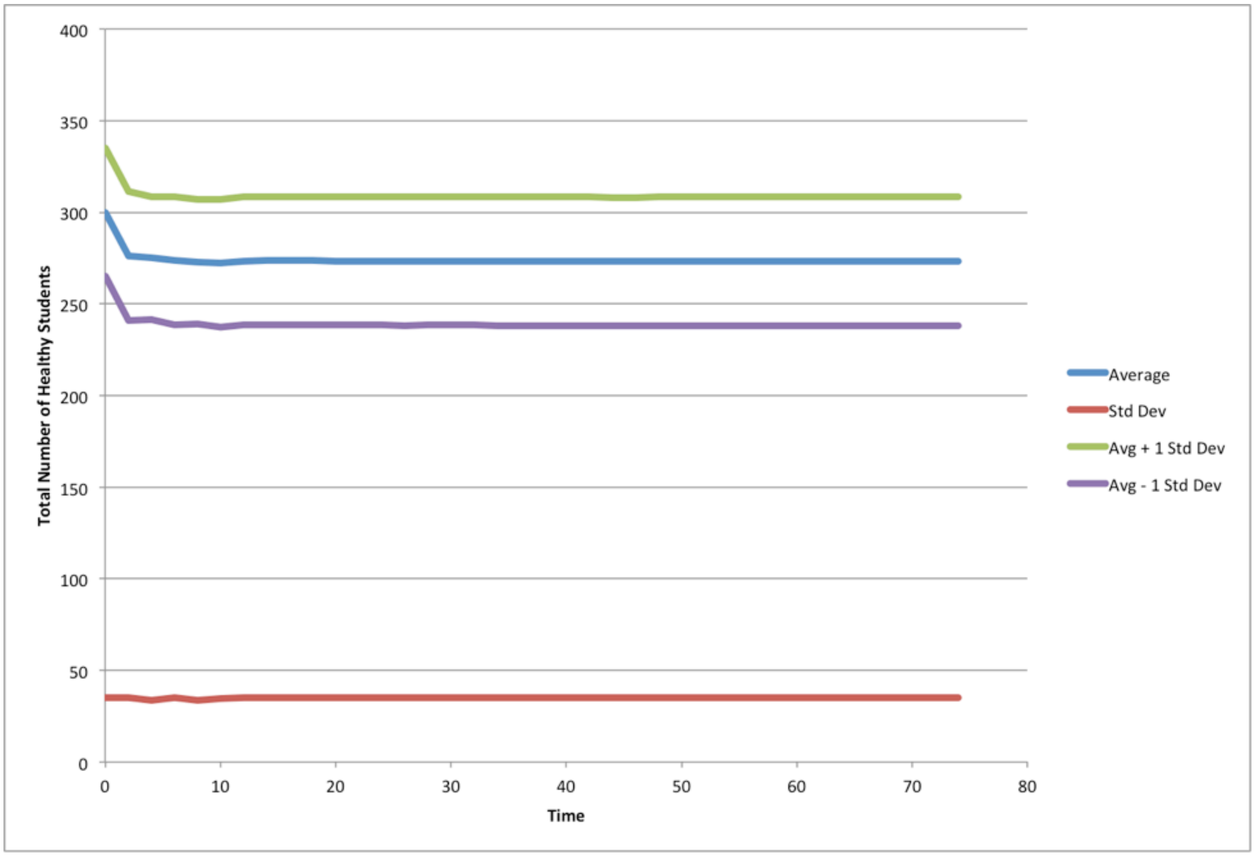


Figure 2 The total number of healthy students as a function of time.

In figure 2 it is showing the total number of healthy students. As you can see, the blue line, or the average, starts high then goes down, but levels off. This is because of the detector. The point when the line starts to level off is when all the agents move to the classroom, or nurses office. When the sick agents move to the nurse's office they are no longer spreading the disease, because they aren't interacting with healthy agents.

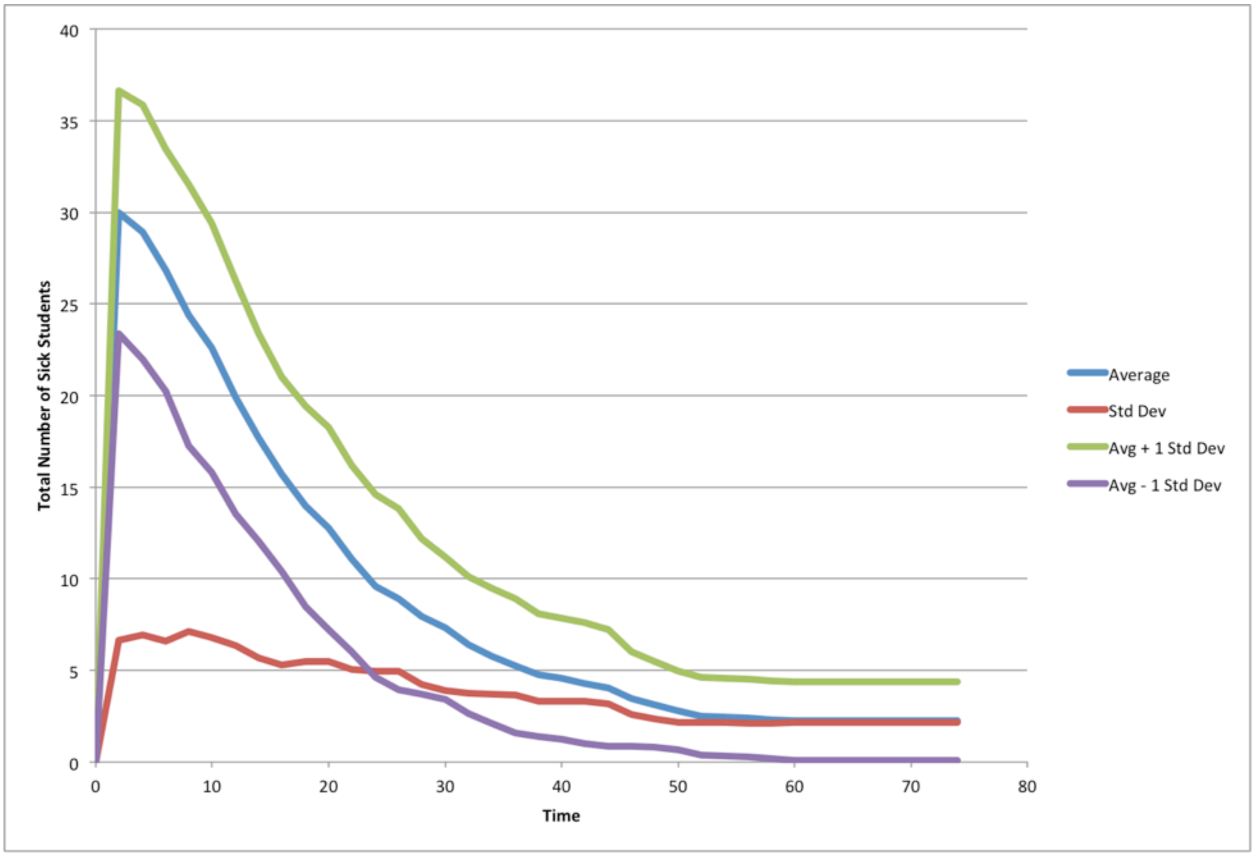


Figure 3 Total number of sick students as a function of time.

In this graph we are showing the total number of sick students. As you can see the blue line, or the average, starts at the bottom and grows, then makes a peak and gradually comes down again. When the line is first going up this is because the agents are out at recess and interacting with other agents. Then the line makes a peak and goes back down, this is because we have moved the agents to their classroom, or the nurse's office if they are sick. So the sick agents aren't colliding with any healthy agents, making them sick.

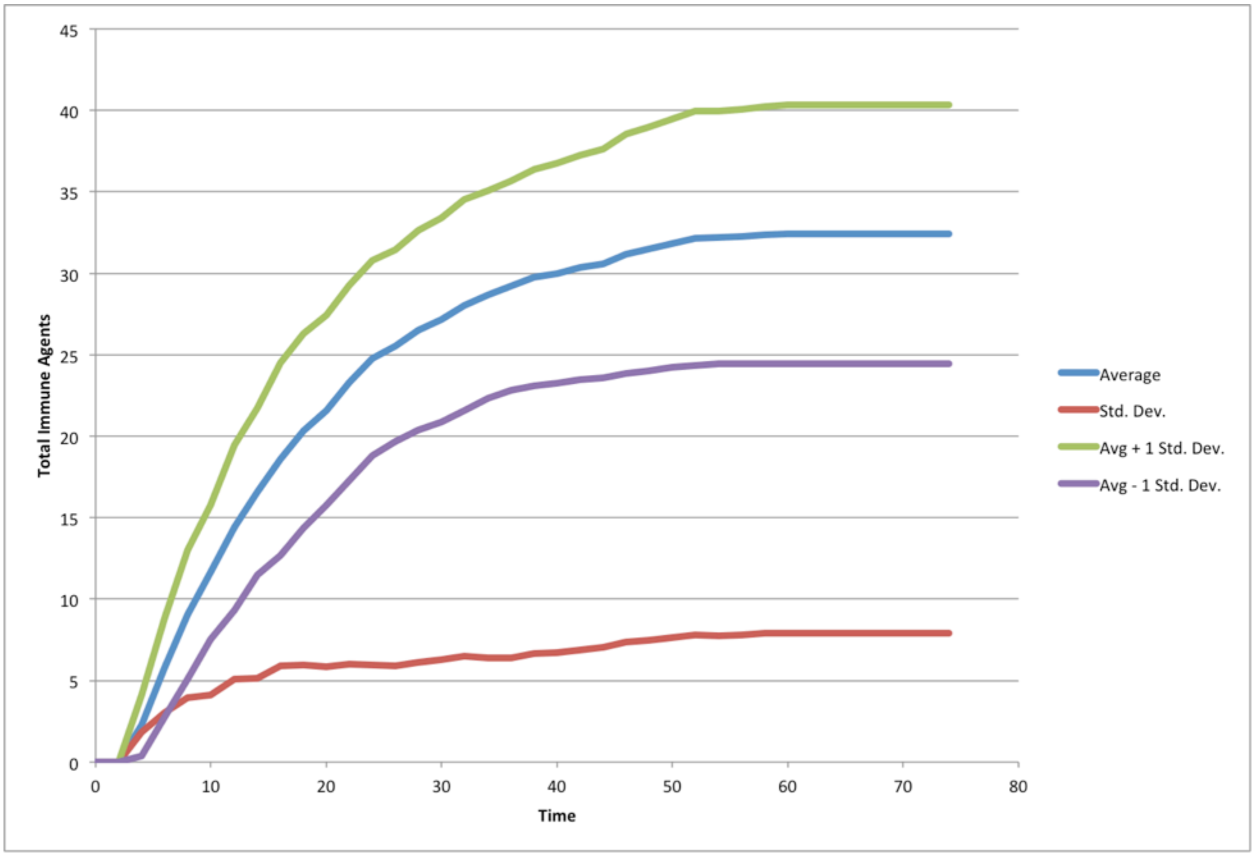


Figure 4 Total number of immune agents.

Figure 4 shows the total number of immune agents. When the blue line, or the average, starts it starts at zero then increases as the sick agents recover. It grows because when the model starts the agents are out at recess, so agents are colliding, getting sick, then recovering and turning immune. So the immune population goes up. Then it starts to level off. This is because we have moved the agents to their classrooms and the sick agents move to the nurse's office. Meaning they are not interacting with other agents, and getting them sick so they can recover and become immune.

Conclusions

Over this school year we have studied and modeled the spread of H1N1. We have interviewed the school nurse to see if our survey results are really realistic. We have tried to make our spaceland as realistic as we could but hit a barrier. It is not possible to put over eight classrooms in our model. Otherwise our logic gets too extensive and we can't work with it. This year we have tried to quantify the uncertainty in our model inputs and their impact on the simulated results. We have also included modeling the effects of using a temperature detector at the school. If the students are sick the detector simulates taking the students temperature, and if they are sick they are quarantined in the nurse's office. The detector has reduced the percentage of sick agents by approximately 50% making a huge impact on our model. We highly recommend using a detector/thermometer in schools. We could minimize the percentage of people who get sick, reducing the number of missed school days for staff and students.

Recommendations For Future Work

We recommend trying to move our model to NetLogo. This might give us more freedom for the spaceland design. We could also model with some percentage of the population having been immunized by having a flu shot. This would make those agents much less likely to go from exposed to sick. But as the immunization wears off, their probability of getting sick could increase.

Appreciation

We would like to thank the following people for helping us and giving us support: Aspen school Nurse, David DeCroix (mentor), Cetin Unal (mentor), Mrs. Zeynep Unal, and our parents.

References

1. Fishman, G. S. (1995). *Monte Carlo: Concepts, Algorithms, and Applications*. New York.
2. McKay, M.D.; Beckman, R.J.; Conover, W.J. (May 1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code" (JSTOR Abstract). *Technometrics* (American Statistical Association) **21** (2): 239–245
3. Brian William, Private Communications, 2011.