Capstone Research Project
Advised by Dr. Raunak and Dr. Olsen

Quantitative Measurements of Model Credibility

Michael Setteducati
BS Computer Science Candidate
Loyola University Maryland
Baltimore Md., 21210

May 8, 2018

***Abstract*—This research project is regarding the development of an online tool for researchers to quantitatively validate simulation models. Background of the field of simulation modeling and the framework is described. Then the process of developing the tool and conducting a user study is outlined, as well as a reflection of the project as a whole.**

I. Background

There is no shortage of published research pertaining to the field of modeling and simulation (M&S). This includes studies regarding the process of developing a simulation model, as well as approaches to the verification and validation of a simulation model [1], [2], [3], [4], [5]. An overview of the M&S field is presented in this section.

Modeling is the process of developing an abstraction of a system which represents the system under study (SUS) accurately in the domain of its application. Although the model is less sophisticated than the actual SUS, a good model will approximate the system well enough for its intended application [6]. In developing a simulation model, there exists an inherent trade-off between realism and simplicity. On one hand, the model should be complex enough that it encompasses the major features of the system. But on the other hand, an overly complex model will be hard to comprehend and maintain [7].

There are four primary types of models regarding its variables and dependencies. *Deterministic models* have fixed input and output values, while *stochastic models* include at least one variable that is probabilistic. *Static models* have no dependency on time. In contrast, *dynamic models* include at least one variable that depends on time [6]. Note that some simulation models may be defined by more than one of these types; for example, a model might depend on time and probability.

In addition to these four types of models, there are two distinct classes of models describing the underlying component that drives the model and its behavior. *Discrete-event simulation (DES)* refers to models that are driven by "event-based changes to system state." For example, a simulation model describing an Emergency Room Department might be driven by patient arrivals and other similar events contained in the model's application domain. In contrast, *agent-based models (ABM)* refer to models that describe "autonomous entities interacting in a spatial environment." A common example is a generic predator-prey model in nature, where predators and prey autonomously interact under the constraints of their environment. It is also plausible to create a hybrid model which incorporates both DES and ABM into a single model [8]. Furthermore, either a DES, ABM, or hybrid model may be deterministic, stochastic, static, dynamic, or some combination of these.

Simulation refers to the physical execution of a model. Simulation is required for studying the behavior of a model, which is representative of the behavior of the actual SUS. Simulation is especially useful for experimenting with parameters of the model, or observing the system's response to some situation of interest when it is infeasible to modify the actual system. It is commonly used to minimize risk when considering changes to an existing system, or the creation of a new system altogether [6].

A *simulation study* incorporates the process of modeling a system, executing simulations of the model, and drawing conclusions based on the results of the model simulations. The success of a simulation study is contingent on following a structured approach in managing the project. Law proposes a seven-step approach for conducting a successful simulation study [7]:

1) The problem is clearly formulated, specific questions are raised, and the scope of the model is defined.
2) High-quality data is collected and the assumptions document is written.
3) A structured walk-through of the assumptions document is performed to ensure that the assumptions being made are acceptable.
4) The model is programmed and verified against the conceptual model.
5) The model is validated against the SUS.
6) Experiments are conducted using the programmed model and the results are analyzed.
7) The findings of the study are documented and presented.

Law's framework mandates returning to a preceding step when the product yielded by a particular step is unsatisfactory; for example, if the assumptions document is not accepted by stakeholders, the project reverts to the previous step and the assumptions document is revised. In a more extreme example, experiments might be conducted but do not resolve the problem, so the project reverts to an earlier step and the scope, assumptions, and/or model are revised [7]. Maria and Balci have independently proposed analogous approaches to conducting a simulation study, which indicates that this method is generally accepted by the M&S community [6], [4].

Essential components of a simulation study are the phases of *verification and validation* (V&V); the process of V&V establishes credibility in the model and therefore trust in the study's results. An overview of the processes of V&V, as described below, is depicted in Fig. 1.

Model verification is the process of "ensuring that the computer program of the computerized model and its implementation are correct" [2]. It is the process of testing the internal structure of the programmed model, and can be thought of as "debugging" the simulation model's code [8]. This process is imperative because it ensures that the program behaves consistently with the conceptual model's design.

Model validation is defined as the "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" [2]. It is essentially the process of testing the simulation model's behavior against the actual behavior of the SUS. In order for a model to be considered valid, it must approximate the behavior of the SUS to a degree that is considered sufficient in the domain of the model's application [8].

Model credibility implies a certain level of trust that the information derived from the simulation model accurately describes the SUS in its application domain [2]. "A simulation model and its results have credibility if the decision-maker and other key project personnel accept them as 'correct'" [7]. The fundamental purpose of V&V is to establish credibility in the model's application domain.

There are a few ways to go about verifying and validating a model regarding the group who performs the V&V process and ultimately makes the decision on the model's validity. The first approach is to have the developers of the model perform V&V throughout development, as well as at its conclusion. The issue with this approach is the fact that the development team is also responsible for testing its work, so if an aspect of the model is overlooked during development, it may also get overlooked during V&V. An alternate approach is to have the users of the model determine its validity, thus allowing key stakeholders in the project to make the important decision regarding the model's validity. Another approach, known as *independent verification and validation* (IV&V), is to have a third party with a thorough understanding of the model and its application domain assess the model's validity. This approach is especially effective in validating large, complex models with multiple development teams because it allows one team to focus solely on V&V throughout development.

Regardless of the approach used, it is preferred to perform validation efforts throughout development, not just at the end, to increase the chance of detecting deficiencies soon after they are introduced. Model deficiencies that go undetected for a long period of time can prove to be costly and time consuming. Furthermore, the group that performs V&V may opt to use a "scoring model" to quantify validation efforts performed on the model. This is generally done by associating scores with each of the various validation techniques employed. The scores for each technique are then aggregated in some way to provide an overall confidence score for the model. However, "this approach is seldom used in practice" [2].

II. Motivation

A. Purpose

Quantifying the validity and credibility of a simulation model is difficult. Nevertheless, it is important to have a metric that describes how rigorously a model has been validated to establish trust in the results derived from experiments using that simulation model.

The concept of quantifying validation is related to the fact that quantification has existed in the software testing community for quite some time [8]. There are well defined and understood metrics in software testing which roughly describe the amount of testing that has been performed on the system. For example, statement coverage and branch coverage describe, in a general sense, how thorough the applied test suite is. Granted, achieving high coverage metrics does not ensure that the system is without defects; however, high coverage does indicate that a large portion of the code has been executed by the test suite, thus, it increases one's confidence in that test suite.

According to Olsen and Raunak, "model validation related activities have generally been categorized as necessarily qualitative in the sense that there is no standard way to quantify the level of confidence gained in the model through validation" [8]. The lack of a standard for describing validation efforts has made it difficult to accurately reproduce models developed for the same system because it is unlikely that unvalidated results will be reproduced by another model. Furthermore, there is no reason to reproduce a model if it has not been thoroughly validated simply because credibility has not been established in that model. A well-documented validation effort is required for people to trust the results of a simulation model [8].

The purpose of this project is to produce a tool for researchers to easily apply the proposed quantitative validation framework to their validation efforts. This tool will also help researchers describe the model's credibility by allowing them to discuss the validation efforts performed with respect to this quantitative framework. "Without a metric to capture and communicate the level of validation performed on a simulation model, and thus a quantified measure of the model credibility, it is difficult to communicate if a model has achieved an appropriate level of confidence for decisions made about the original system based on its findings" [8].

Olsen and Raunak's proposed framework takes quantitative measurements of individual validation tasks (ideally performed

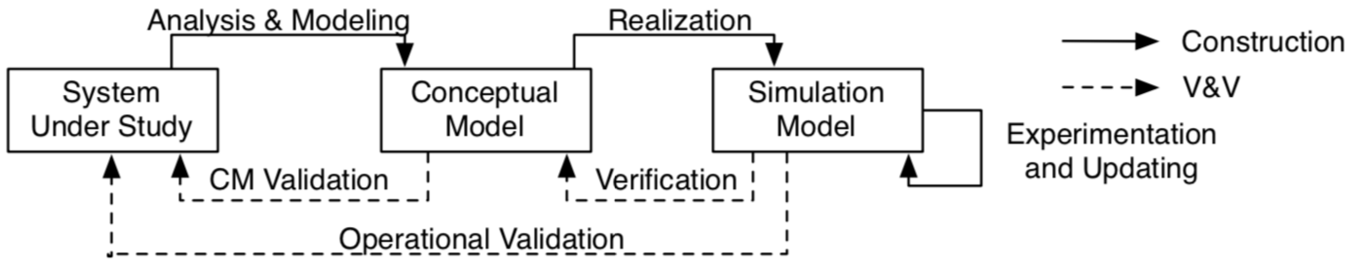

Fig. 1. High Level Overview of the process of developing a simulation modeling. Note that verification ensures that the simulation model is consistent with the conceptual model, whereas validation ensures that the conceptual model accurately describes the system under study. Graphic is from [8].

throughout the development process) and aggregates these measurements into an overall confidence metric representative of the total validation performed on the model. There are three primary areas of validation defined in this framework: structural, behavioral, and data validation. *Structural validation* refers to validating the internal structure of the model. *Behavioral validation* refers to validating the external behaviors manifested by the model. *Data validation* refers to validating the data used to execute the model. All three components are essential to thoroughly validating a system. There are subtle nuances in the way that each type of validation is performed, so a confidence score is independently calculated for each validation type. The overall confidence metric is then calculated by aggregating these three confidence scores [8]. Overall, this tool will eliminate the process of going through the framework manually to calculate the overall confidence metric in the hope that it motivates researchers to use the framework and more thoroughly validate their model.

B. Broader Impacts

This online quantitative validation tool could potentially be used regularly by researchers in the M&S community. This includes a broad variety of fields that utilize M&S, such as engineering, biology, psychology, and others. The tool could help researchers talk specifically *and quantitatively* about the validation activities performed on their model. This will benefit other researchers attempting to reproduce or improve some already developed model. This will also give users a greater (or lesser) degree of confidence in the model depending on its overall confidence score. Ideally, this framework will ultimately lead to the development of more credible models, and in turn, more accurate research and more informed decisions.

III. Plan of Work

The outcome of this research project will be an online tool for quantitatively validating a given simulation model; however, there are a few prerequisites essential to developing this application with a high degree of quality. The first prerequisite is knowledge of simulation modeling in general, as well as the role V&V plays throughout the modeling process. This background information might provide beneficial insight regarding potential applications of the tool. The second prerequisite is familiarity with various validation techniques, including each technique's purpose and applicable use cases. Awareness of common validation techniques should promote improved realization of the tool's intended functionality. The final prerequisite is a thorough understanding of the quantitative validation framework proposed by Olsen and Raunak [8]. Understanding this framework is essential to the project because it will facilitate verification that the tool's behavior is consistent with the proposed technique.

Once these prerequisites are met, development of this quantitative validation tool will begin. The software development process will begin by collecting requirements and designing the system in conjunction with Dr. Olsen and Dr. Raunak. The system will then be developed and tested following an agile methodology, thus allowing modifications of system requirements and design throughout the process.

Regarding system design, the validation framework will be written in Python, and the Flask web micro-framework will be used to serve the application via a RESTful API. The front-end will be written using Angular, however it will be possible for users to interface directly with the API if they choose.

Once the tool is developed and thoroughly tested, experiments will be done to assess the usability and functionality of the application. A user study will be conducted by having participants enter elements and techniques of a simulation model and attempt to calculate the validation confidence. This will help assess the usability of the application. A case study will also be conducted by having someone in the modeling and simulation field use the tool to validate their model. This will help assess the functionality of the application because this person should be familiar with the work that goes into validating a model.

A. Challenges

The most challenging component of this project is likely going to be designing the quantitative validation tool in such a way that the underlying framework's high degree of flexibility is emergent in the tool's functionality. The proposed framework does not define specific validation techniques that must be performed; instead, only techniques relevant to the application domain contribute to the overall confidence metric. The framework also allows users to modify the weight associated with a given validation task. Therefore, the tool must dynamically handle this inherent flexibility. [8]

The corresponding solution is to place an emphasis on the design stage of development, especially when developing the core algorithm that computes the overall confidence metric. This algorithm must account for any combination of validation tasks deemed relevant to the model application's domain. By using an appropriate software architecture and best coding practices, the produced software solution will support the dynamic nature of the underlying framework.

Additionally, the tool must be intuitive and easy to use. It should be simple for researchers to input validation performed, as well as customize their associated weights. This requirement will be addressed via the user study during the experiments phase of the project. The tool will also store user simulation studies so that validation efforts may be added and updated throughout the study so that users do not have to continuously import and export their models.

Another challenge of building this tool will be ensuring that it is secure. It is imperative that user data is not compromised; the data may be sensitive, or this tool may be the primary store of validation efforts on some study. Furthermore, this tool should not compromise the "cs.loyola.edu" server in any way.

B. *Resources*

The collection of resources used in this project is comprised of academic papers regarding the field of M&S. The primary resource used is the book chapter written by Olsen and Raunak which proposes the quantitative validation framework that is the underlying basis of the validation tool being developed [8]. This resource will be explicitly followed during the design and development of the tool to ensure the program's accuracy. There are also numerous secondary resources referenced in this project's documentation regarding V&V of simulation models and other topics in the field of M&S.

Another useful resource will of course be the Internet. The Internet will be the source of documentation for Angular and Flask, and will be heavily relied upon to develop this tool. Furthermore, the Internet is a source of additional M&S and V&V literature that will be required to write the related works section.

IV. Timeline

Planned timeline as originally proposed at the start of this project in January, 2018. The actual timeline throughout this project generally followed this proposed timeline.

Week	Dates	Task
1	1/15 - 1/20	Discuss general idea of topic
2	1/21 - 1/27	Discuss plan for semester and the concept of simulation
3	1/28 - 2/3	Read simulation and validation papers by Law and Sargent
4	2/4 - 2/10	Read V&V papers by Kleijnen, Balci, and Wang
5	2/11 - 2/17	Complete Background and Motivation document App development - Requirements and Design
6	2/18 - 2/24	App development - Design
7	2/25 - 3/3	Complete Related Works document App development - Implementation
(break)	3/4 - 3/10	App development - Implementation
8	3/11 - 3/17	App development - Implementation and Testing
9	3/18 - 3/24	App development - Testing
10	3/25 - 3/31	App development - Deployment Complete Conference paper (potentially)
11	4/1 - 4/7	Conduct experiments using app
12	4/8 - 4/14	Conduct experiments using app
13	4/15 - 4/21	Experimental results write-up
14	4/22 - 4/28	Complete final research report draft
15	4/29 - 5/5	Submit final research report

V. About Simulation Studies

In conducting a simulation study, it is imperative to the success of the study that a structured and well-defined approach is followed. Many researchers have considered the best way to organize a simulation study and there exists a generally agreed upon method to conducting a successful study [1], [4], [6]. Note that although much of this research has slight variations regarding the best way to conduct a simulation study, the underlying structure is basically identical.

Step one is to *formulate the problem* being addressed by the study [1], [4], [6]. This includes defining the objectives and requirements of the study, as well as the scope of the model and specific questions that the model will aim to answer [1], [6].

Step two is to *collect information on the system* and form an assumptions document [1], [4], [6]. This includes analyzing the system under study's (SUS) environment and interdependencies, as well as appropriate input variables for the system model [4], [6]. Additionally, creating a detailed assumptions document is vital for defining the scope of the system so it can be referred to in later phases of the study [1].

Step three is to *develop the model* [4], [6]. This includes creating a conceptual model as well as actually programming the model. The assumptions document is referred to here because the model is based on the assumptions that have already been made [1]. Balci describes modeling as "an artful balances of opposites" because the model must include essential elements of the system but also exclude unnecessary details that would add to the model's complexity [4]. It is vital to understand that a model is an abstraction of a system created for a specific purpose in accordance with the assumptions document, so it is unnecessary and actually detrimental to attempt to include all of the details of the system in the model [4].

Step four is to *verify and validate the model*, referred to as "V&V" of the model [1], [6]. This process includes running different tests and analyses on the model to compare expected model output to it actual output [1], [6]. It is a the most crucial step in establishing model credibility. This phase is the focus of this research, and will be discussed in more detail later.

Step five is to *design and conduct experiments* on the model that are within the model's application domain [1], [4], [6]. This step is typically the the entire purpose of the simulation study, to experiment with the system and make decisions based on the experimental results. The assumptions document should be referred to here to ensure that the experiments do not violate any of the assumptions that have been made and that the experiments are within the model's domain [6].

Step six is to *document and present the results* of the study [1], [6], [4]. This step is important because it gives other researchers information about the study and its findings. The results should be presented with respect to the original intended purpose of the system, in accordance with the assumptions document [4].

It is important to note that the cycle is inherently cyclical in nature. When an error occurs during some phase, the cycle reverts to an earlier stage to correct the error. Additionally, a study may go through multiple passes of the cycle, improving the product of the previous pass each time [4].

Overall, it is important to follow an organized and repeatable approach to conducting a simulation study to optimize efficiency and effectiveness of the study [1]. The study should also be well documented so that future researchers can model related studies off of it and ideally improve the model and experimental results [4]. The quantitative validation tool that is the subject of this research project aims to increase the amount of documentation that is done during a simulation study, especially documentation regarding validation efforts, by giving researchers a way to quantify their efforts in their documentation. This quantification and increased documentation should lead to building more credible models.

VI. Verification and Validation

The phase of V&V is the most vital part of a simulation study in establishing the credibility of a simulation model [4]. As Olsen and Raunak explain, "establishing model credibility thus includes that the model (a) is internally consistent with no known errors (verification) and (b) mimics the SUS's behavior to a level of confidence necessary for making the model useful for its intended application (validation)" [8]. If model credibility is not established, then the results of the experiments on the model cannot be trusted, so confidence in the model is essential in trusting its results [8].

A. Verification and Validation Techniques

There are a multitude of verification and validation techniques that have been studied and documented by researchers, and in practice some are more common than others. With respect to the quantitative framework proposed by Olsen and Raunak, there exists three primary categories of validation techniques: structural, behavioral, and data [8]. Olsen and Raunak define specific techniques that have been deemed most essential to establishing model credibility for each category; however, it is important to note that the framework does allow for alternative techniques to be conducted and applied to any of these categories.

1) Structural Validation: This type of validation refers to testing the internal structure of the model in comparison to the system's organization, interactions and processes [8]. There are five primary techniques defined by Olsen and Raunak as the most relevant techniques to throughly validating a model: parameter verification, dimensional consistency, structure verification, extreme condition, and boundary adequacy.

Parameter verification ensures that the internal and input parameters of the model are appropriate and accurate with regard to the SUS and the model's application domain [2], [8]. This technique should be conducted with respect to the assumptions document because the parameters of the model should be within the scope of the model and not infringe on any of the prior assumptions made.

Dimensional consistency evaluates structural assumptions made to ensure that the model's structure is consistent with the SUS and the model's application domain, and should be run in conjunction with parameter verification [8].

Structure verification ensures that the structure of the model is also apparent in the SUS with regard to organization, decision-making, and assumptions. This test should be conducted by both the modeler and an expert in the field [8].

Extreme condition ensures that the model can appropriately handle extreme values and scenarios; if it cannot, there is likely an underlying issue with the current structure of the model that does not accurately represent the SUS [8].

Boundary adequacy evaluates the level of abstraction that has been chosen with regard to the actual system. It incorporates the abstractions, assumptions, and data used to design the model, and is defined by Olsen and Raunak as the most important test to ensure structural validity [8].

2) Behavioral Validation: This type of validation involves validating emergent, external manifestations of the behavior of internal entities of the system [8]. The relevant behavioral validation techniques defined by Olsen and Raunak are: animation, degenerate tests, internal validity, Turing tests, face validation, sensitivity analysis, metamorphic validation, model comparison, trace data, and results validation.

Animation testing is done by creating a visualization of the system model to evaluate how the model changes over time and ensure that this visualization seems to be an accurate representation of the system [2].

Degenerate tests ensures that the "degeneracy of the model" is appropriate with regard to input and internal parameters. This definition is more easily understood through the following example: "does the average number in the queue of a single server continue to increase over time when the arrival rate is larger than the service rate?" [2].

Internal validity involves running a stochastic model multiple times to determine the amount of variability in the system [2], [4]. By estimating the amount of internal variability in the system, a large amount of variability should call into question the validity of the system's behavior [2].

Turing tests are conducted by presenting field experts with real system output as well as the output of the model to see if the expert can discern which output was produced by the actual system and which was produced by the model [4], [2]. If the output of the SUS and model are similar, the confidence in the model is greatly improved [4].

Face validation is similar to Turing tests in that field experts subjectively compare the model's behavior to the behavior of the SUS to determine if the model behaves reasonably [2], [4].

Sensitivity analysis involves systemically modifying the input and internal parameters of the model to determine the parameter's effect on the system and evaluate this effect with respect to the effect it would have on the SUS [2], [4]. If the effect of changing a parameter drastically has an unexplainable effect on the output, it reveals inconsistencies in the structure of the model [4].

Metamorphic validation ensures that the SUS has appropriate data to effectively validate the model using other validation techniques [4].

Model comparison is comparing the output of the model being developed against other model's that have already been validated [2].

Trace data is following the behavior of specific entities in the system to ensure that they act logically and appropriately as they do in the SUS [2]. This is an effective way of validating individual entities' behavior in the system, as opposed to the behavior of the overall system.

Results validation involves analyzing the results of the simulation study with respect to expected results and results of related studies that have already been conducted [2].

3) Data Validation: This type of validation includes validating the data used to build the model and data used to run the model, but in the context of this research refers specifically to the data used to run the model [8]. The two most important techniques for validating data are goodness of fit tests and face validity.

Goodness of fit tests involve comparing the distribution of model output with the expected output distribution of the SUS [2]. This ensures that the data has a normal distribution, especially when validating a stochastic model.

Face validity, as was previously defined, is having a field expert compare the model input data to the input data of the SUS [2], [4]. The difference in this definition is that the field expert is analyzing the data, not the behavior, as was the case in the previous definition.

4) Application of these Techniques: All of the aforementioned techniques are commonly used to validate elements of a simulation model by researchers in the field. Thus, they are all present in the online tool as available techniques that might have been applied. Researchers will be able to input the results of the application of these techniques to validate specific elements of their model in the online tool. These results will then be included in the confidence score of the relevant validation category (structural, behavioral, or data), as well as the model's overall confidence score.

B. Best practices for Increasing Model Validity

In addition to running validation techniques on the model, it is important to conduct the simulation study in accordance with accepted best practices to improve the model's credibility. These practices should be well documented throughout the study so that other researchers who review the study get a sense for the manner in which the study was conducted. Some of the most documented practices are detailed below.

First, it is important to collect good data on the system and have conversations with field experts about the SUS, including its behaviors and data that will be used [7], [1]. Additionally, modelers should be familiar with previous studies regarding the same system to further improve their understanding of the system being modeled [1]. This familiarity enables researchers to model the system as accurately as possible, and ideally improve the overall confidence of the model.

Next, it is imperative to communicate with the managers of the study on a regular basis to have a clear idea of the desired final product. This also gives the managers an idea of the progress that is being made and established a good line of communication between the managers and modelers. Furthermore, the manager's decision regarding the model's credibility is crucial, so by keeping them in the loop throughout the study they are more likely to deem the model credible [7]. The validation tool will help improve this line of communication by simplifying the process of producing a report of validation efforts that have taken place.

It is also very important to perform a structured walk-through of the assumptions document with all relevant stakeholders so that everyone understands what is within the scope of the model. This keeps everyone on the same page, and increases model validity by ensuring that only what is within the scope of the model is actually developed [7].

Finally, it is important to use quantitative techniques to validate the model as opposed to subjective tests because it is easier to measure quantitative validation [7]. This is the point of this quantitative validation framework: to better understand how much validation has been done and thus how credible the model is.

VII. Quantitative Model Validation Framework

The validation process of a simulation study is a crucial step in obtaining model credibility. The Modeling and Simulation field does not have a well established method for quantitatively validating a simulation model; nonetheless, it would be beneficial to simulation studies and the field as a whole if there was a way to quantitatively describe the validation efforts performed on a model. Research conducted by Dr. Olsen and Dr. Raunak has led to a proposed framework for doing so [8], [9], [10], [11].

There are four properties of a model as proposed by the quantitative model validation framework: purpose, structure, behavior, and data. The purpose regards the motivation for building the model, as well as the scope of the model, and defines what questions might be asked of it. This property is not quantitatively incorporated into the model, however the structure and behavior of the model are based on this purpose so it is important that it is well defined. The structure of the model describes the internal organization and interactions of the entities in the system. The behavior of the model describes the external manifestations of behaviors of internal entities in the system. The data property regards the data used to execute the model [8].

The quantitative validation framework breaks down elements and techniques by these three properties, structural validation, behavioral validation, and data validation. A confidence score is computed for each of these properties based on the validation that has been done on this property of the model. Finally, the structural confidence, behavioral confidence, and data confidence are aggregated to compute an overall model confidence according to the following formula:

$$mc = 0.5bc + 0.3sc + 0.2dc$$

where *mc* denotes model confidence, *bc* denotes behavioral confidence, *sc* denotes structural confidence, and *dc* denotes data confidence. These weights for each property have been assigned based on Dr. Olsen and Dr. Raunak's research regarding the importance of each of these properties as they pertain to the overall model, however they may be altered at the user's discretion [8].

To calculate the confidence of the structural, behavioral, and data validation properties, elements in the model that require validation are identified and validation techniques relevant to these elements are identified and applied. For structural validation, elements are either considered "Creation Data" or "Other," each of which has specific techniques that must be applied. For behavioral and data validation, it is up to the user to determine which techniques are applicable to each element.

Then, the success of each technique pertaining to a given element must be indicated. In the case of structural validation, each technique is assigned a success score between 0 and 1 with respect to each element. For behavioral and data validation, each technique is scored on the binary basis of whether or not it was successfully completed.

Furthermore, each validation technique has an associated Maximum Confidence which indicates the degree to which it validates the model as compared to other techniques. The Maximum Confidence for a technique is the same across all elements the technique was applied to. Dr. Olsen and Dr. Raunak have suggested Maximum Confidence levels for each technique, however these numbers may be changed at the discretion of the user.

The confidence score for each property is aggregated based on the results of these validation techniques on each element. The formula for computing the confidence for a given property takes into account the success of the technique, as well as the Maximum Confidence of that technique. For full details of the mathematical calculations for each property, refer to [8].

VIII. Quantitative Validation Tool

We propose that an online web application will simplify the process of calculating the structural, behavioral, and data validation coverage, as well as the overall model confidence. An online tool will make it easier to organize validation efforts and run the calculations. This tool will increase the likelihood that simulation studies employ this quantitative validation framework during the validation process, and report their validation level in publications.

The tool will have all of the necessary information regarding the framework to make it easy for users to understand the tool and necessary input. First, the home page will describe the complete framework for reference. Additionally, information will be dynamically displayed based next to the on-screen controls that the user is interacting with to walk the user through the process of inputting data.

At a high-level, the tool will allow users to identify structural, behavioral, and data elements that need to be validation. It will then allow them to add relevant techniques and indicate the success of that technique, as shown in Fig. 2. Ultimately, it will display a report of all of these techniques, as well as the confidence scores for each validation property and the overall model confidence, as show in Fig. A-11.

The in-depth list of functional requirements for the tool, as well as references to associated screen shots in Appendix A, are as follows:

- The system shall allow users to input validated elements of the model. (Fig. 2, Fig. A-8, Fig. A-9)
- The system shall allow users to input validation techniques performed on elements of the model. (Fig. 2, Fig. A-8, Fig. A-9)
- The system shall allow users to add validation techniques not stored by default. (Fig. A-10)
- The system shall allow users to modify the potential confidence of a validation technique. (Fig. A-10)
- The system shall allow users to modify the weights of the validation categories (structural, behavioral, and data) that are used to calculate the overall confidence. (Fig. A-10)
- The system shall produce an overall confidence metric for the model given the validation techniques performed. (Fig. A-11)
- The system shall produce reports of the validation techniques performed on each element, and the model's validation coverage metrics (behavioral, structural, data, and overall). (Fig. A-11)
- The system shall allow users to store the model being tested in a JSON file. (Fig. A-6)
- The system shall allow users to upload previously created models from a JSON file. (Fig. A-6)
- The system shall allow users to interface directly with the API. (Fig. A-14)
- The system shall allow users to create an account and login to it. (Fig. A-12)
- The system shall allow logged-in users to save projects, as well as retrieve and update previously created models that are saved in the system. (Fig. A-13, Fig. A-6)

A. System Design

The system uses a client-server architecture such that the server hosts the application and handles the computations

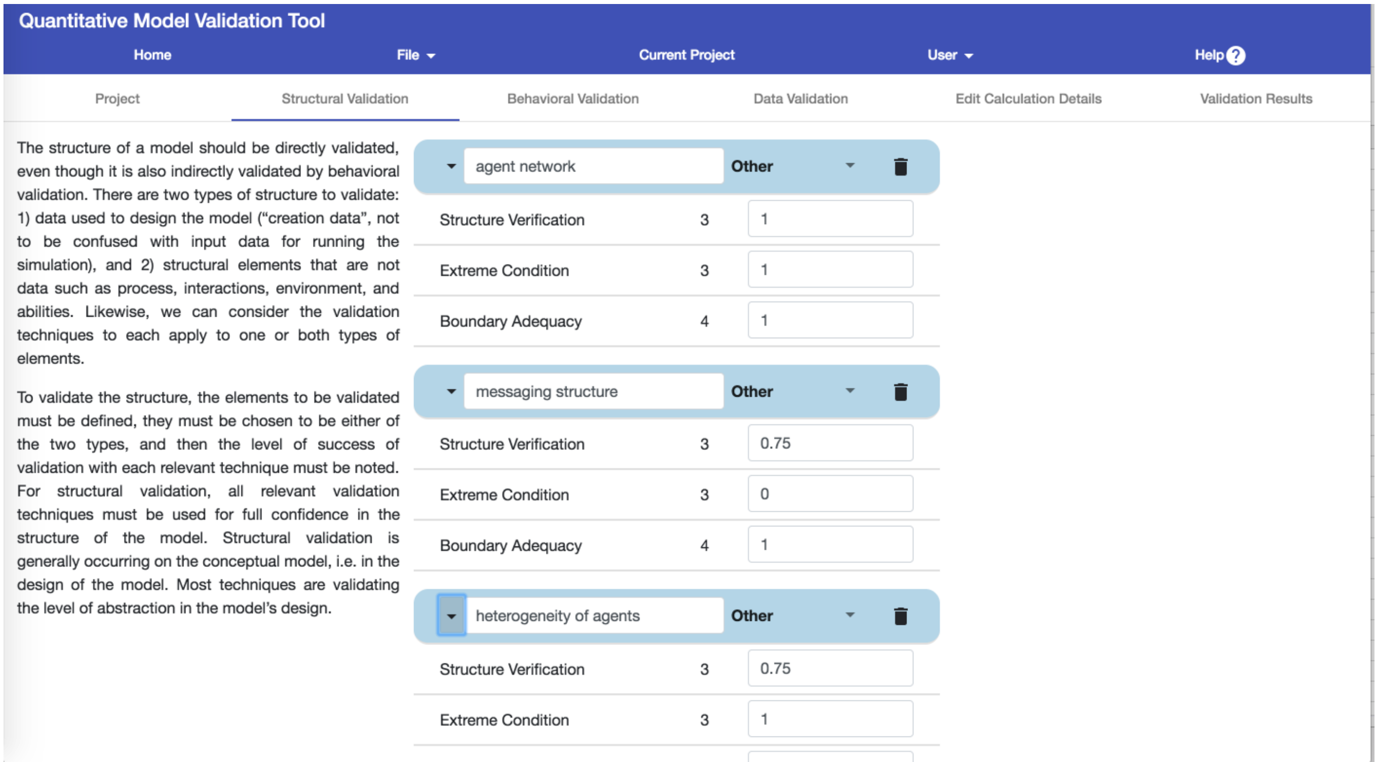

Fig. 2. Example of adding a three Structural Elements of type Other with varying success scores between 0 and 1.

required by the framework. The server serves the UI of the web application to the client, which makes requests to the API hosted on the server to run the calculation and respond with the confidence scores of the given model. The API also serves requests related to authentication and user accounts, and is able to retrieve models that are stored in the database once a user is logged-in.

Python is the primary language that is used on the back-end. Flask is a web micro-framework that is used to handle requests to the server. The flask API takes requests which contain a model (stored in JSON), and uses a python module to run the quantitative model calculations, then the API creates a response and sends it back the client. The flask API also deals with user logins; flask has a built-in component for user authentication. Finally, flask API retrieves models stored on the server that a user has created.

User authentication data and stored models are stored in a NoSQL database powered by MongoDB. MongoDB natively stores JSON data, making it an ideal choice for storing these validation models because validation requests containing model data are already stored in JSON format.

The front-end is written using the Angular web framework, so the code is primarily written in TypeScript (and HTML and CSS). This allows the front-end application to exist on a single web page. Furthermore, data is updated synchronously on throughout the components of the application. Angular services are used to store and manipulate the data, and the angular components access the data from these services. These angular services are also responsible for making requests and handling responses to the back-end API. For example, when a user inputs a validated element on the structural validation component, this element automatically appears in the component that summarizes the model in a table because both components are using the same data stored in the same service.

B. Software Tests

Software tests will be required to ensure the accuracy of the calculation, as well as ensuring that the front-end and back-end respond appropriately in various situations.

1) Quantitative Framework Algorithm Tests: Unit tests are required to verify the correctness of the algorithm. They test the functions of the sub-modules, such as adding a new element or technique, as well as testing for the correctness of the confidence scores against some model that has been quantitatively validated by hand. Example models used in [8] are used as test data.

2) Front-end Angular Tests: Jasmine is a JavaScript testing framework for Angular applications that is used to verify the behavior of the Angular components and services. These tests ensure that various components are inserted or removed when they are expected to have been.

3) Back-end API Tests: The Flask application uses the unittest module that is built in to python to test the responses to various requests to the api. Tests will be written to ensure that properly formatted models are properly validated, and that improperly formatted requests are responded to appropriately. Tests will also be written to ensure authentication when attempting to access models that are stored in the database.

IX. Experiments

The experiments component of this research is primarily regarding the usability of the quantitative validation tool. The tool must be easy to use so that simulation researchers are able to seamlessly use it throughout a simulation study. The tool must also be easy to understand so that people who are not familiar with this quantitative validation framework are able to use the tool effectively and understand what exactly is being validated. We thus plan to conduct both a user study and a case study to validate the usability of the tool.

A. User Study

A user study will be conducted during which Loyola CS upper-level undergraduates are asked to complete various tasks using the online tool. For example, they will be asked to add various elements and techniques to the model and calculate the overall model confidence. After going through a variety of tasks, they will then be asked to fill out a survey regarding how usable they found the tool to be. This will give us feedback on the tool's UI and allow us to gauge how easy it is to use the tool.

The study will take place between April 23, 2018 and May 4, 2018. We are planning to have 5-10 Loyola CS undergraduate students participate in the study. We will be collecting data on each task that they perform regarding how easy they found it to complete that task using a survey conducted at the conclusion of the final task.

The following are the list of tasks that a participant will perform during the study:

1) Overview: Find information in the application about the Quantitative Model Validation Framework and read it to get a sense of what this tool does.

2) Model Creation: You have a simulation model regarding an Emergency Department and you would like to quantify the validation efforts performed to this point. Create a new project for this model.

3) Adding Structural Elements: You have identified that structural elements of the model include Emergency Department Workflow, Treatment Beds, and Patient Arrival Rate. You have determined that Emergency Department Workflow is of type Other, and all the applicable techniques were all 100% successful. Next, Treatment Beds is of type Other, and Structural Validation and Boundary Adequacy Tests were 100% successful, but Extreme Condition Tests were not successful. Finally, Patient Arrival Rate is of type Other, and Parameter Verification, Dimensional Consistency, and Boundary Adequacy Tests were all 100% successful. Input this information into your project.

4) Adding Behavioral Elements: You have identified the following behavioral elements: Average Wait Time for Bed, Doctor Utility Rate, and Nurse Utility Rate. Average Wait Time for a Bed has a weight of 100% relative to the other elements, whereas Doctor Utility Rate and Nurse Utility Rate have half the weight of Average Wait Time. For each element, the following techniques are applicable: Degenerate Tests, Face Validation, Sensitivity Analysis, Model Comparison, Metamorphic Validation and Results Validation. However, for each element, only the following techniques were successfully applied: Degenerate Tests, Face Validation, Model Comparison, Metamorphic Validation, and Results Validation. Input this information into your project.

5) Adding Data Elements: The following two elements were identified as validatable data elements: Number of Doctors and Number of Nurses. They both have a weight of 50%, and both successfully applied Face Validation, which was the only applicable technique.

6) Computing Validation Coverage: Compute the structural validation coverage, behavioral validation coverage, data validation coverage, and overall model confidence of your current project

7) Export Model: Export this model to a JSON file and save it to your Desktop.

8) Import Model: Import the model stored in the file called "GossipModel.json" from your desktop. What is the overall model confidence of the Gossip Model?

9) Editing Model: Add an element to the structural validation property of the model called Agent relationship, which represents the type of relationship between the various entities in the model (e.g. close friend). The relevant techniques for Agent relationship are Structure Verification, Extreme Condition Tests, and Boundary Adequacy Tests. Structure verification was validated 100%, but Extreme Condition and Boundary Adequacy Tests were only validated 75%. What is the confidence score of this modified model?

10) Adding a new Technique: Add Continuity Testing as a new behavioral validation technique with a maximum confidence of 6.

B. Case Study

A case study will be conducted in which a person in the field of modeling and simulation is given access to the tool to use on one or more of the simulation models that they are working on. He/she will then give us feedback on the tool, and we will use this feedback to improve the UI. This case study is an ideal way to test the functionality of the application because this person will have a better understanding of the tool's application than someone who does not know much about simulation models (as is the case in the user study). At least one Ph.D. student from University of Texas Arlington will participate in the case study.

X. Experimental Results

A. User Study

The user study was ultimately conducted with four total participants. As previously mentioned, they were given a series of tasks to complete and the completion time of each task was recorded. This data is shown in Fig. 3. At the conclusion of these tasks, a survey was filled out to indicate how much difficulty they had with each task. This data is shown in Fig. 4. Note that each task had an average completion rate of 100%.

First, the Overview task took some users relatively long to complete. However this is because the task involved reading information about the framework and some of the participants more thoroughly read the information on the home page. As

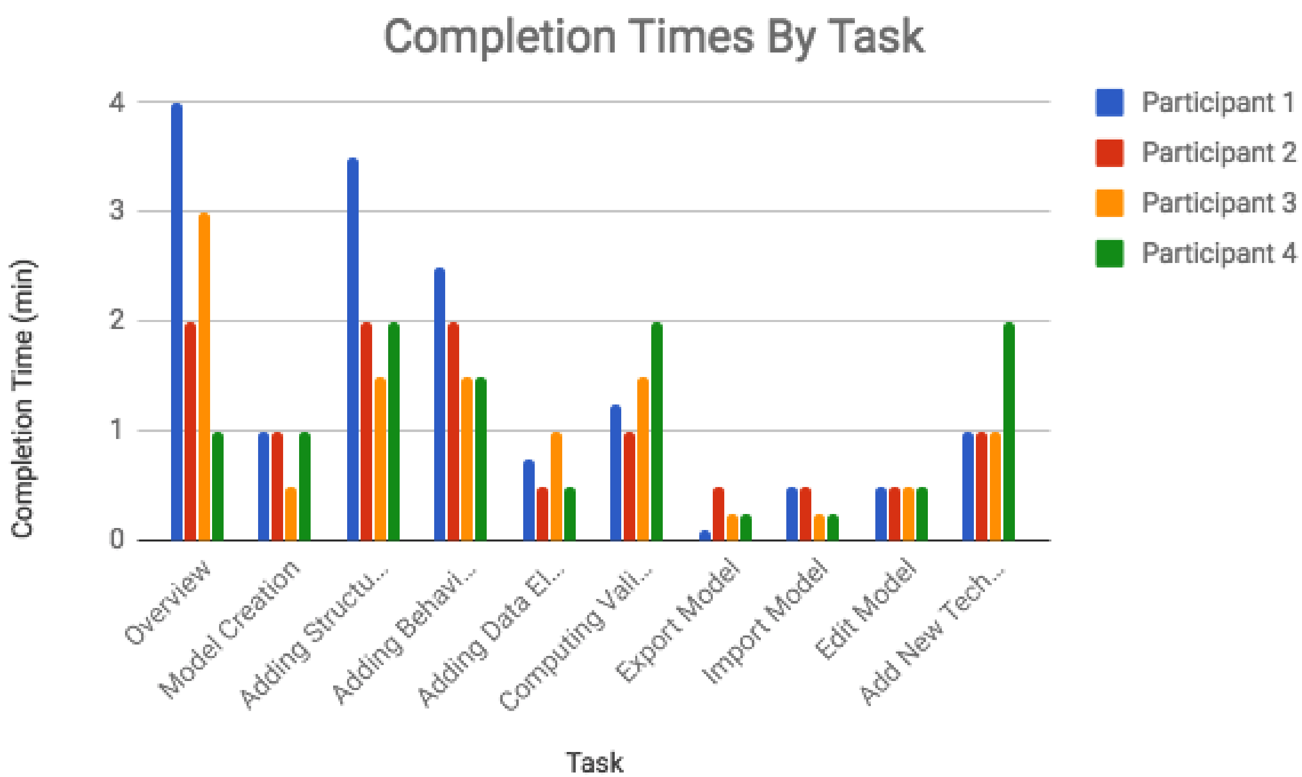

Fig. 3. This chart represents the amount of time it took each participant to complete each task. Note that the Overview task took longer for some participants because they more thoroughly read through the description of the framework on the Home page.

a result, there is no useful information that comes out of this task, except for the fact that users found it easy to find, which is indicated by the average difficulty rating of 1.

Next, the completion times for adding structural, behavioral, and data elements tended to get faster as the user got more comfortable adding tasks. This is why adding Structural Elements took longer than adding Behavioral and Data Elements. Overall, users generally indicated that adding elements for all three validation properties was not difficult.

Next, the completion times for Computing Validation Coverage seemed higher than it should be, especially because all the task involved is technically clicking one button that says "Validate Model." Difficulty ratings, as well as verbal feedback from users, indicate that the button was hard to find because it is on the bottom of a table and is generally not on the screen until the user scrolls to the bottom.

Finally, users generally had trouble adding a new technique to the project. This is partially because the Edit Calculation Page has some text with a disclaimer that the user must accept before making changes. The text was centered which made it hard to read, so users generally skipped past it before coming back to it and realizing that this is where techniques can be added. However, the intention of creating this disclaimer page is to deter users from making changes to the calculation. The default weights and maximum confidence scores are the result of a series of studies conducted by Dr. Olsen and Dr. Raunak, and should not be changed unless the researcher has a good understanding of what they mean. Ultimately, this finding actually confirmed that the design does indeed make it more difficult for users to change the application, as it was intended to do.

One other product of the research study was verbal feedback about the Context Relevant Guidelines, which are the text boxes that dynamically change depending on what control the user has selected when inputting elements and techniques. Users generally did not realize that they were changing, and did not bother to read the paragraph on the side of the page. These guidelines are intended to help users go through the process of inputting data, however they do not serve that purpose if users are unlikely to read them.

B. *Case Study*

Unfortunately, the case study turned out to be unsuccessful because the student that we had recruited to participate dropped out at the last minute, and there was no time to find another participant. It would have been beneficial to get feedback from him because he is significantly more familiar with simulation models than the participants in the user study, but it was ultimately out of our control.

C. *Conclusions*

The data and feedback collected during the experiments led to some minor changes in the application. First, another

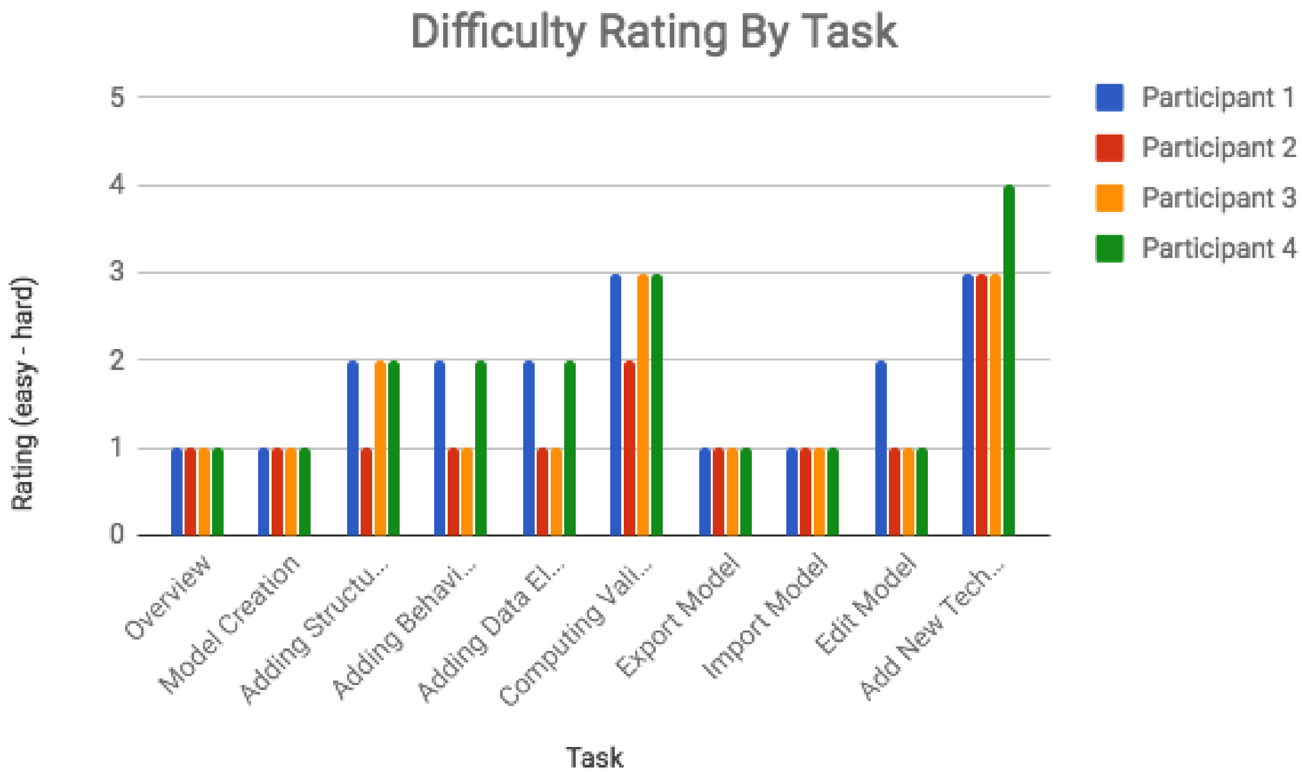

Fig. 4. This chart represents the difficulty rating that each participant assigned to each task in the survey at the end.

"Validate Model" button was added to the Validation Results page at the top to make it easier to find. Next, the disclaimer text on the Edit Calculation page was centered, but is now left justified to make it easier to read. However, the user still must accept the disclaimer to not make the page as easily accessible. Finally, the context relevant guidelines will be converted to bullet points to make it easier to read. There was also no title for these guidelines, so it was not obvious to the user that the guidelines are changing. Therefore, titles will also be added indicating what guidelines are currently displaying. Overall, the user study confirmed that adding elements and assigning them techniques and scores is relatively straightforward, and the tasks that were not completed as quickly led to usability enhancements in the application.

XI. Future Work

The first piece of future work that could be done regarding this research project is a more thorough user study. Although a user study was conducted, it was not completed to the greatest extent that it could have been. Ideally, there would have been more participants and consequently more data to analyze and draw conclusions from. In the future, a more complete user study could be conducted with more participants to further explore the usability of the application.

Next, the case study was not successfully completed but could still be done in the future. The case study would have been a valuable mechanism for getting feedback from someone who is familiar with the field of simulation modeling. Although it was unsuccessful, this feedback would still be very useful to improving the tool moving forward.

Nonetheless, once the tool is used by people in the community, the Loyola Validation team should get feedback on it and be able to make enhancements based on this feedback. Early adopters of the tool are likely to be connections of Dr. Olsen and Dr. Raunak, which will enable them to directly communicate regarding ways the tool could be improved. This may not be an official case study, however the feedback will likely serve the same purpose. Although this is not as ideal as making usability modifications before it is deployed, it is of course never too late to improve a piece of software.

The other primary piece of future work will be actually maintaining the application. This includes ensuring that the tool runs as expected and does not experience downtime. Dr. Olsen and Dr. Raunak will be primarily maintaining the application, so a code walkthrough will be held so they know exactly how the software is built.

Maintaining and improving the software should be relatively straightforward because Angular applications are inherently modular. Various Angular components each pertain to a some specific part of the application, for example there is one component of the application the results table. Angular services are responsible for storing the data and communicating with the components, for example there is one service called Validation Service that handles storing the model's elements and applied techniques and making requests to the API to validate the model. The components that handle inputting elements and techniques send data to this service and the service keeps the data consistent throughout components of the application. Overall, this structure is ideal for simplifying the processes of maintaining and enhancing the application.

A. Impact

This tool will hopefully have a great impact in research conducted in the simulation modeling community. As previously stated, the hope is that researchers start to use the tool to employ the quantitative validation framework in their models. This will help increase model credibility and simplify the process documenting validation efforts.

In an effort to increase awareness about the tool in the modeling and simulation field, Dr. Olsen and Dr. Raunak will be be presenting this tool at the 2018 Summer Simulation Conference in Bordeaux, France. This is an opportune time to show the tool to prevalent researchers in the field with the hope that they will start using it to validate their models. This will also be an opportunity to get feedback on the tool from target users.

XII. Reflection

Overall, this research project has been a great learning opportunity. I now have a good grasp of the field of modeling and simulation, and also learned a lot about full stack web development. I had never used Angular or Flask before, and had never deployed an application to a web server. Now I can confidently say that I have a solid understanding of how to develop and deploy a full stack application. I also now know Angular quite well, and hope to use this framework in the future in a professional environment.

Developing and deploying the application is what I found most rewarding about the project as a whole. It was quite satisfying to go onto the internet after it was deployed to find the application that I had been running on localhost for the past month and a half.

Of course, this development process was not without bumps in the road. There were numerous occasions when I could not figure something out, for example why was the data was not updating on different components, or why did the API call not go through. However, struggling through these errors by browsing a seemingly endless stream of stack overflow threads is how you learn something new. And once I figured something out, I did not run into that problem again because I now knew the right way to do it.

Another issue that I faced was the lack of users to participate in the user study, as well as the student who was going to complete the case study dropping out at the last minute. I dealt with this by going forward with what I had control over and trying to make the best out of the data I was able to collect. I ultimately did receive some useful feedback from the user study and incorporated these suggestions into the application.

To future students, I would suggest starting experiments such as a user study once the earliest version of the software is ready to go to production. I waited to polish the software off a little more before I attempted to start it, and I think I would have had more participation if I did the user study with an earlier version of the application.

Overall, the process of completing a research project, in this case a project that is mostly centered on software development, was both educational and rewarding. My biggest takeaway will be the full stack development experience that I gained throughout the process; however, I also value the knowledge I have gained about simulation models. Simulation modeling is quite interesting and I did not realize how large the community is and how much work has gone into the field.

Feel free to go check out the application at http://validation2018.cs.loyola.edu/tool/.

References

[1] A. Law, "How to Build Valid and Credible Simulation Models," in *Proc. 2008 Winter Simulation Conf.*, pp. 39–47.
[2] R. Sargent, "Verification and Validation of Simulation Models," in *Proc. 2010 Winter Simulation Conf.*, pp. 166–183.
[3] J. Kleijnen, "Verification and validation of simulation models," *European Journal of Operational Research*, vol. 82, pp. 145–162, Apr. 1995.
[4] O. Balci, "How to Assess the Acceptability and Credibility of Simulation Results," in *Proc. 1989 Winter Simulation Conf.*, pp. 62–71.
[5] Z. Wang, "Selecting Verification and Validation Techniques for Simulation Projects: A Planning and Tailoring Strategy," in *Proc. 2013 Winter Simulation Conf.*, pp. 1233–1244.
[6] A. Maria, "Introduction to Modeling and Simulation," in *Proc. 1997 Winter Simulation Conf.*, pp. 7–13.
[7] A. Law, "Building Valid, Credible, and Appropriately Detailed Simulation Models," in *Simulation Modeling and Analysis*, 5th ed. United States: McGraw-Hill, 2014, pp. 246–268.
[8] M. Olsen and M. Raunak, "Quantitative Measurements of Model Credibility," in *Model Engineering for Simulation*. Elsevier, 2018, to be published.
[9] ——, "A Method for Quantified Confidence of DEVS Validation," in *Proc. SpringSim TMS/DEVS*, Apr. 2015.
[10] M. Raunak and M. Olsen, "Quantifying Validation of Discrete Event Models," in *Proc. Winter Simulation Conf.*, Dec. 2014.
[11] M. Olsen and M. Raunak, "A Framework for Simulation Validation Coverage," in *Proc. Winter Simulation Conf.*, 2013.

Appendix A
Validation Tool Screenshots

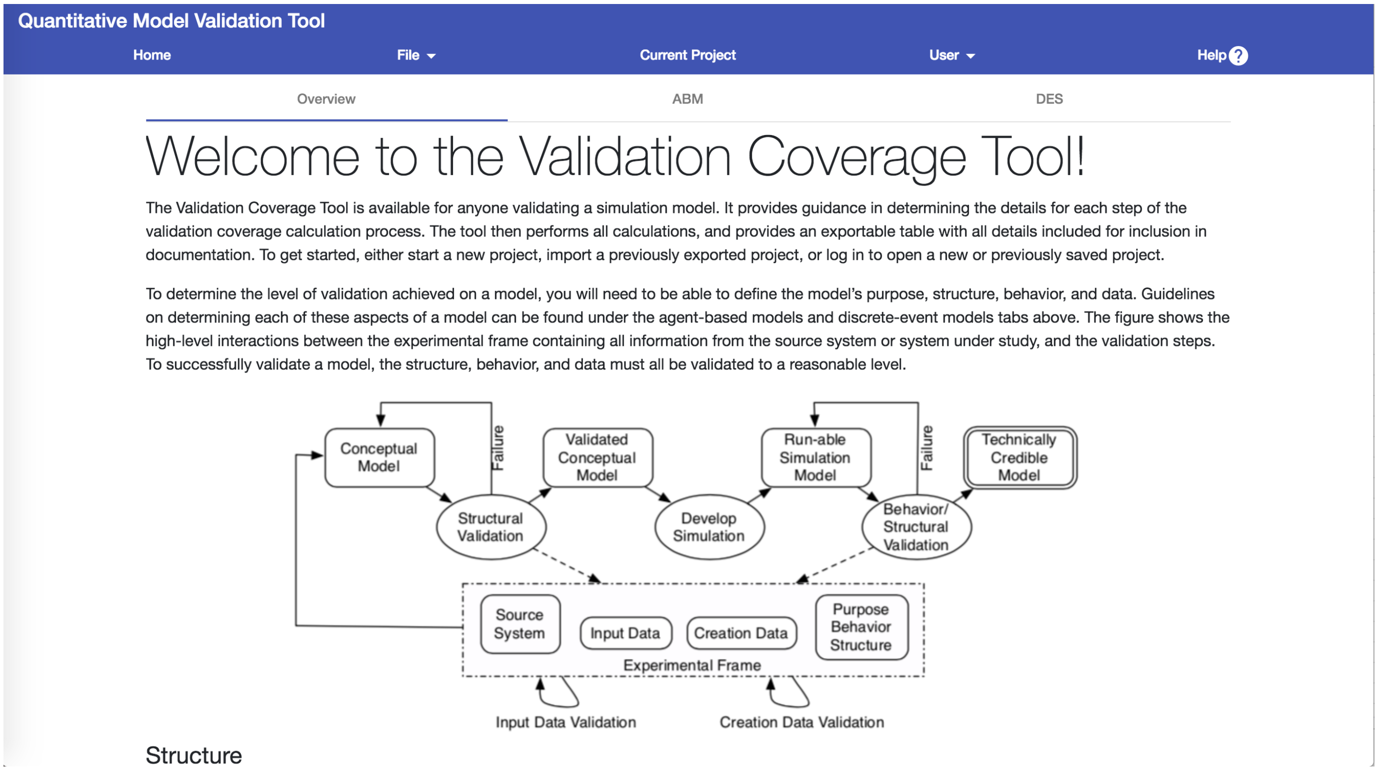

Fig. 5. The Home screen contains information regarding the framework and how to use it.

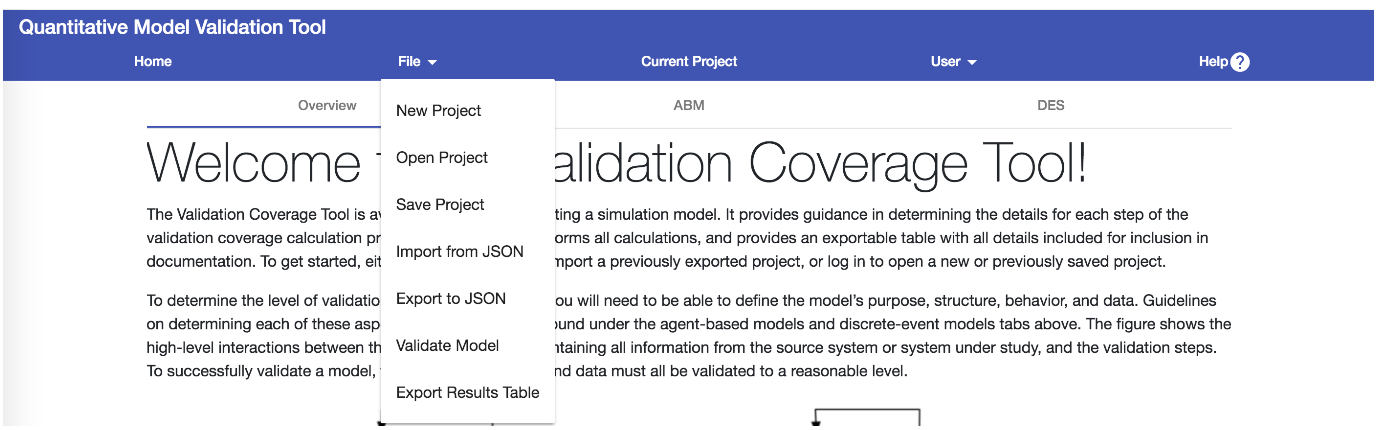

Fig. 6. The File menu allows users to create a new project, open a saved project, save the current project, import from a JSON file, export to a JSON file, validate the current model, and export the results table.

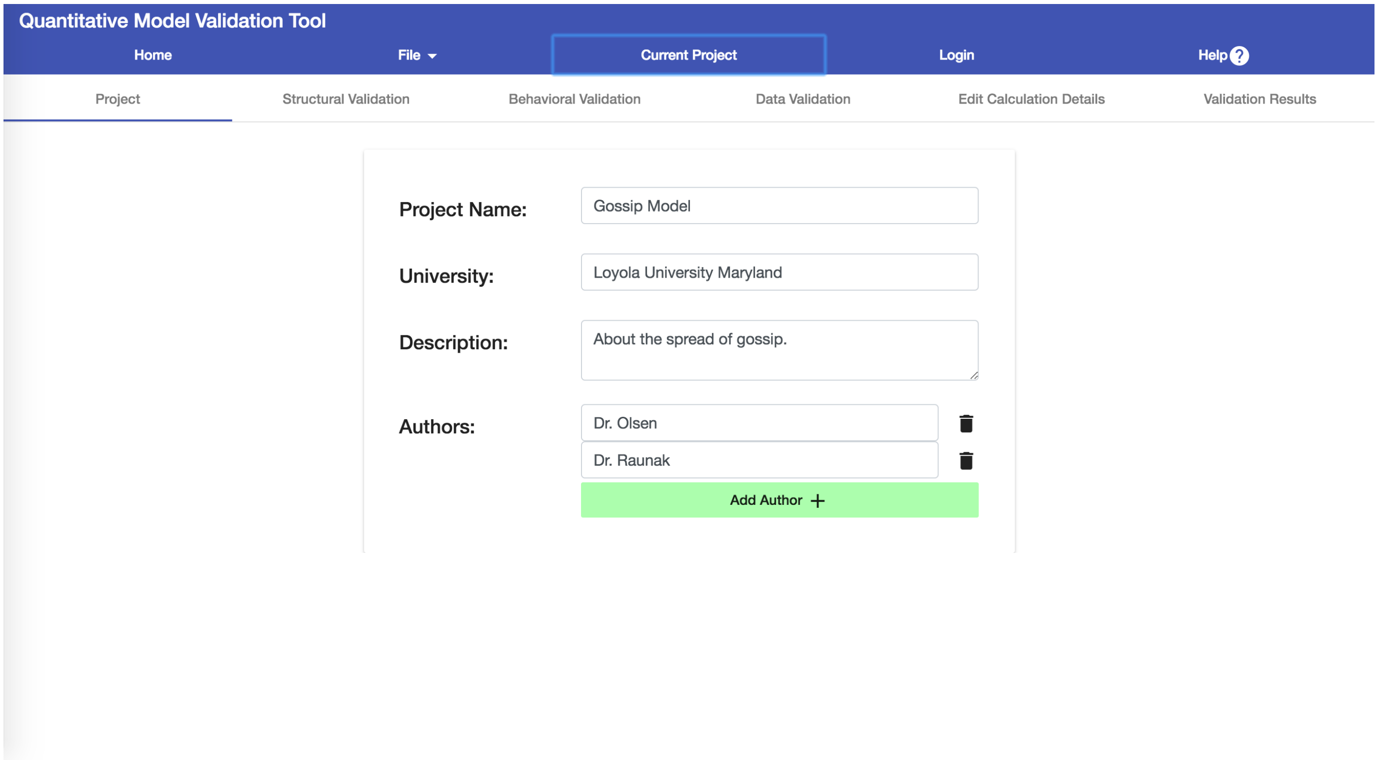

Fig. 7. Project Overview page displays information regarding the model being validated.

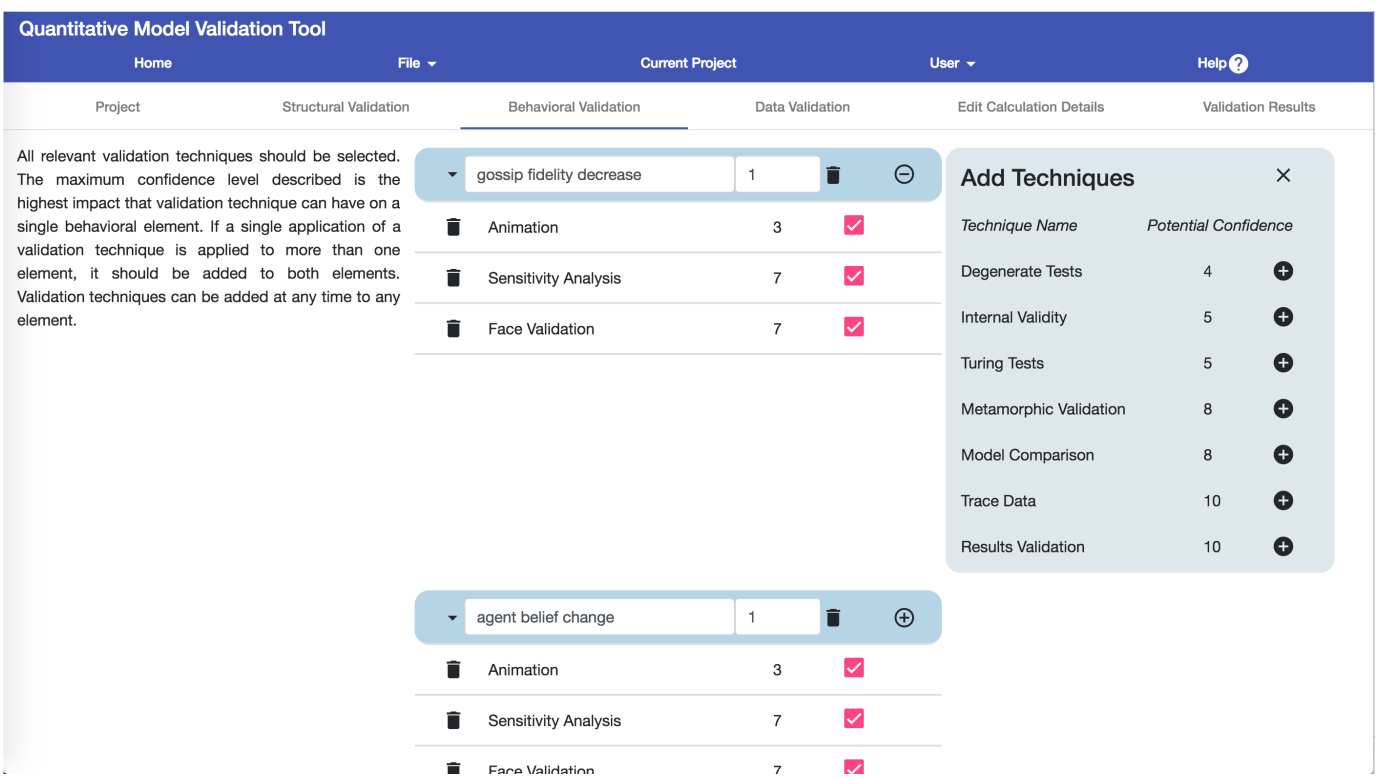

Fig. 8. Behavioral Validation tab allows user to input behavioral elements and relevant behavioral techniques.

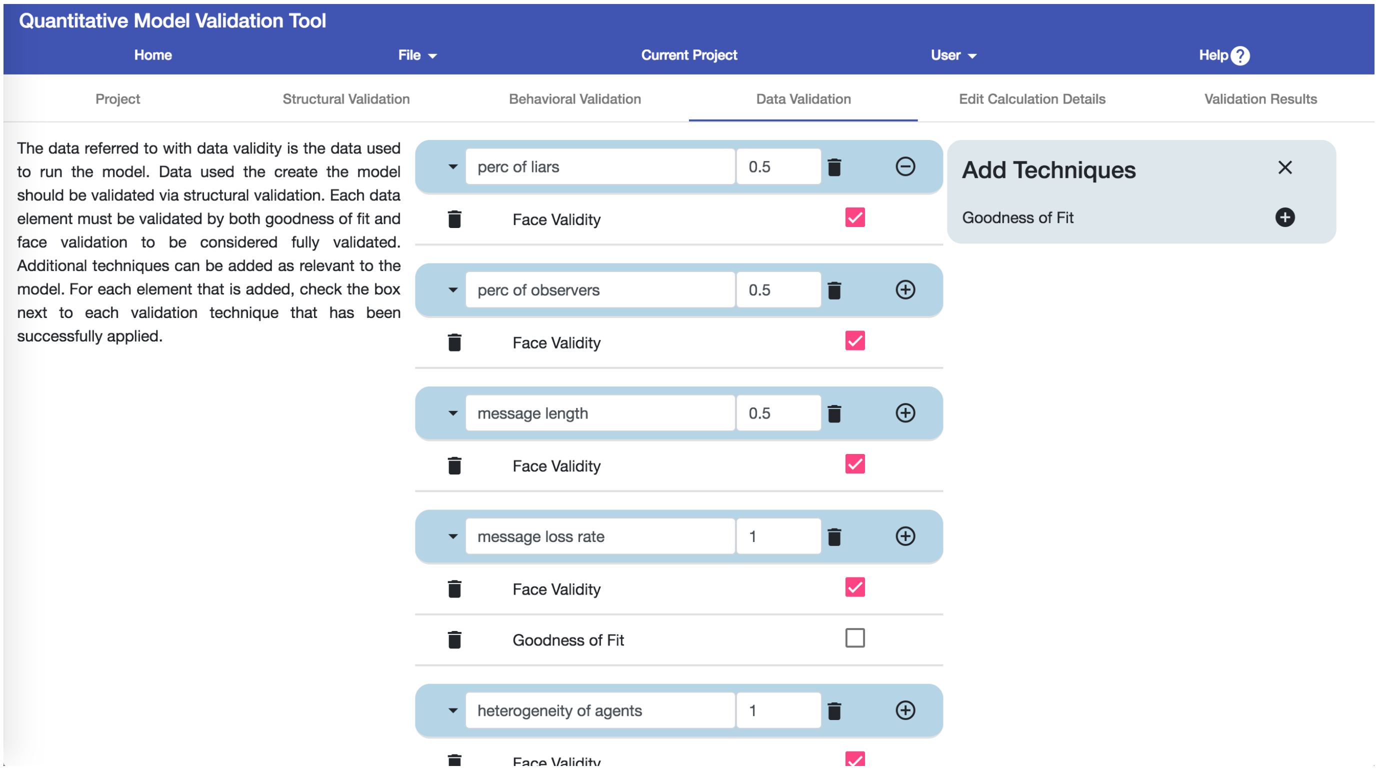

Fig. 9. Data Validation tab allows user to input data elements and relevant data techniques.

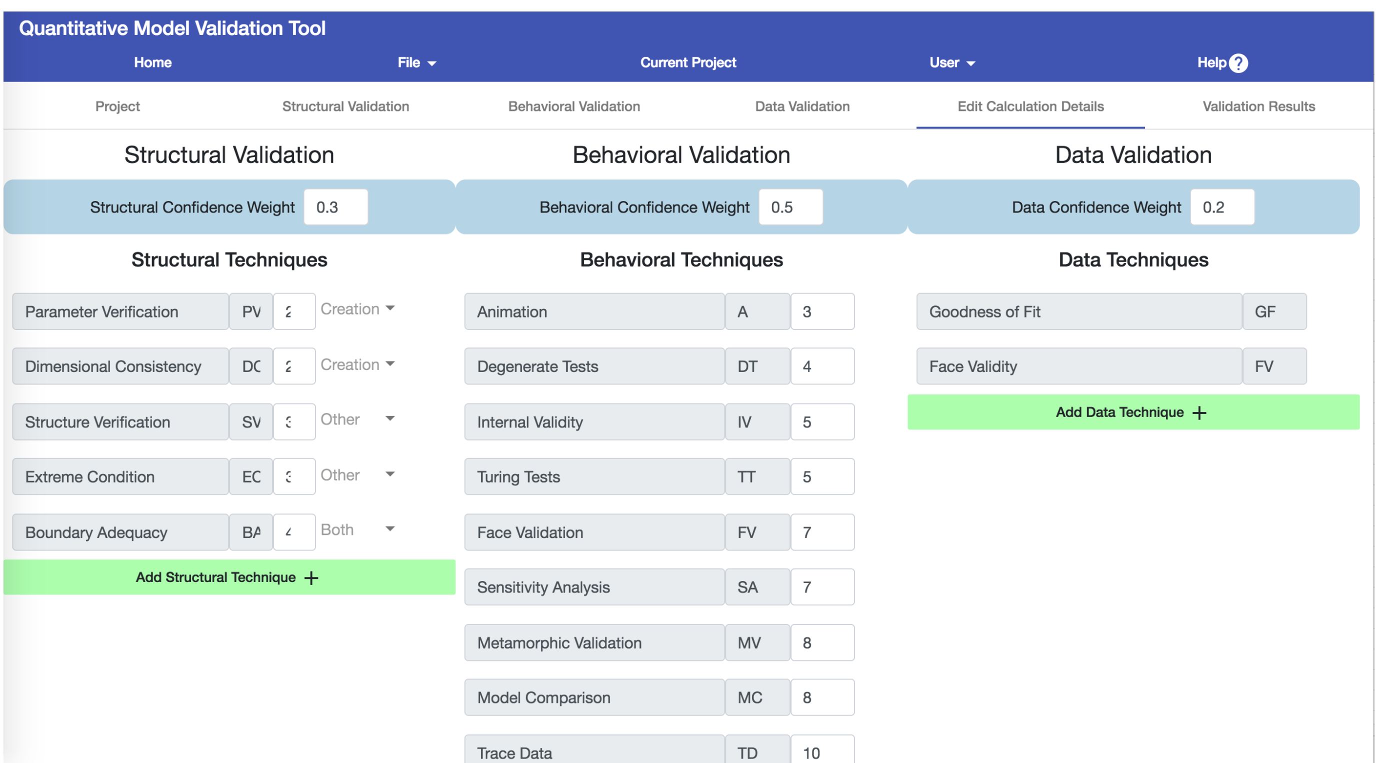

Fig. 10. Edit Calculation tab allows user to alter the weights of each property, the maximum confidence of each technique, and add new techniques.

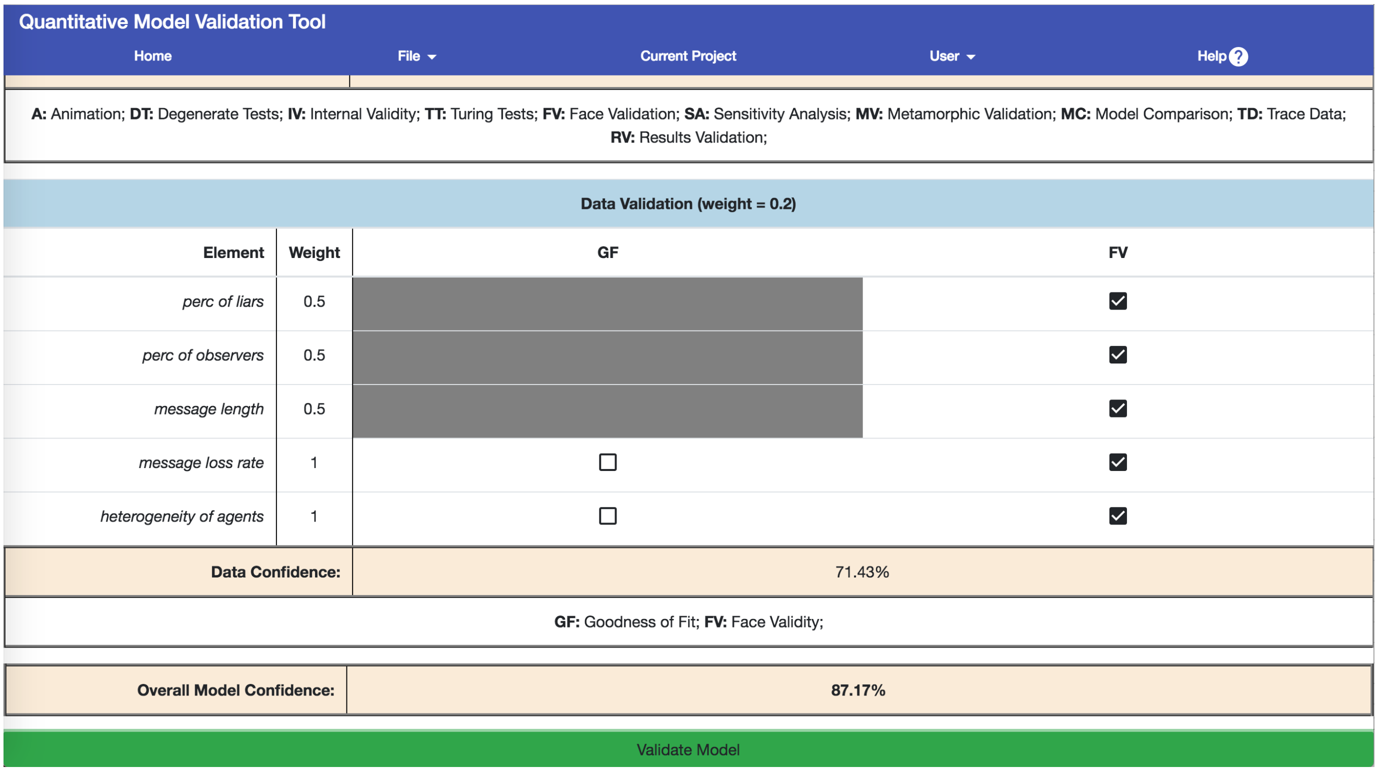

Fig. 11. Validation Results Table displaying Data Elements, as well as the Data Confidence and Overall Model Confidence. Note that Structural Elements and Behavioral Elements are shown in tables above this on the page.

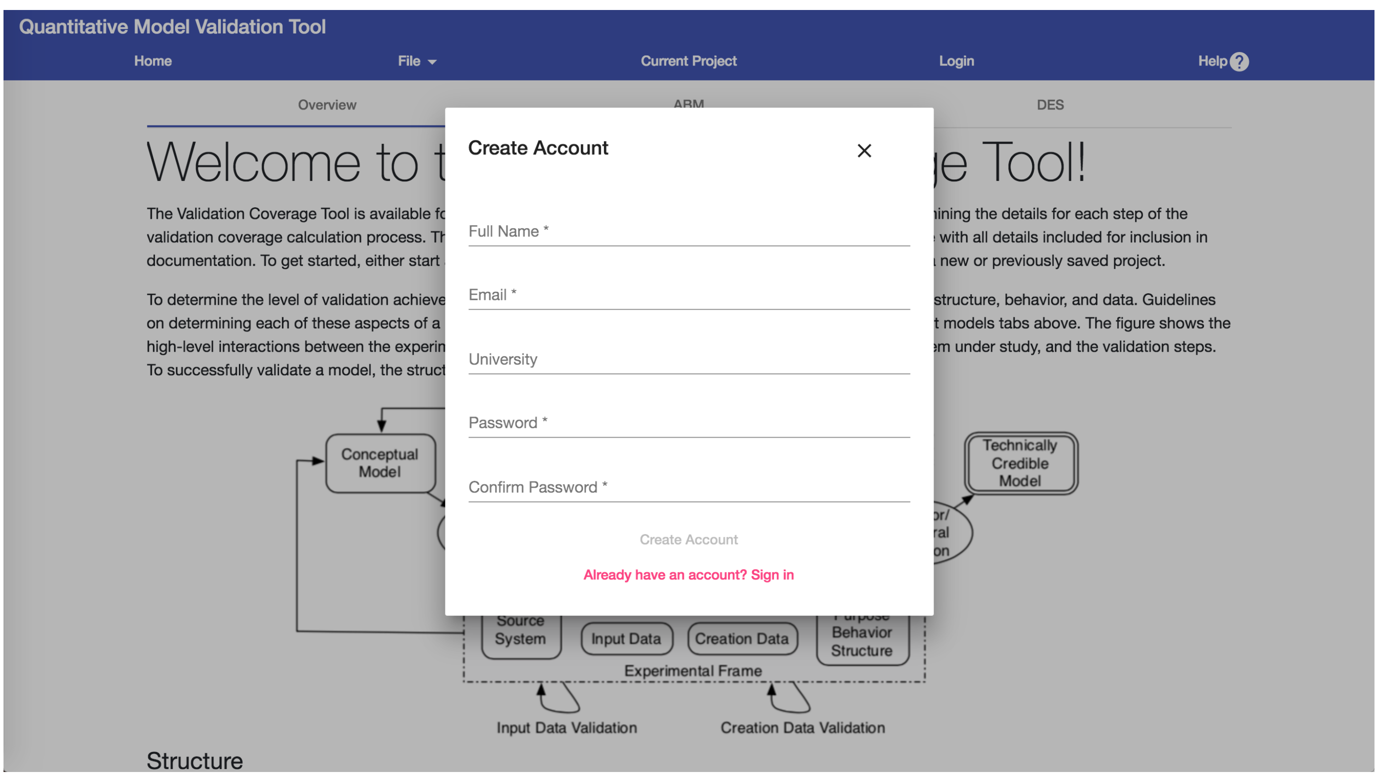

Fig. 12. This form allows users to create a new account.

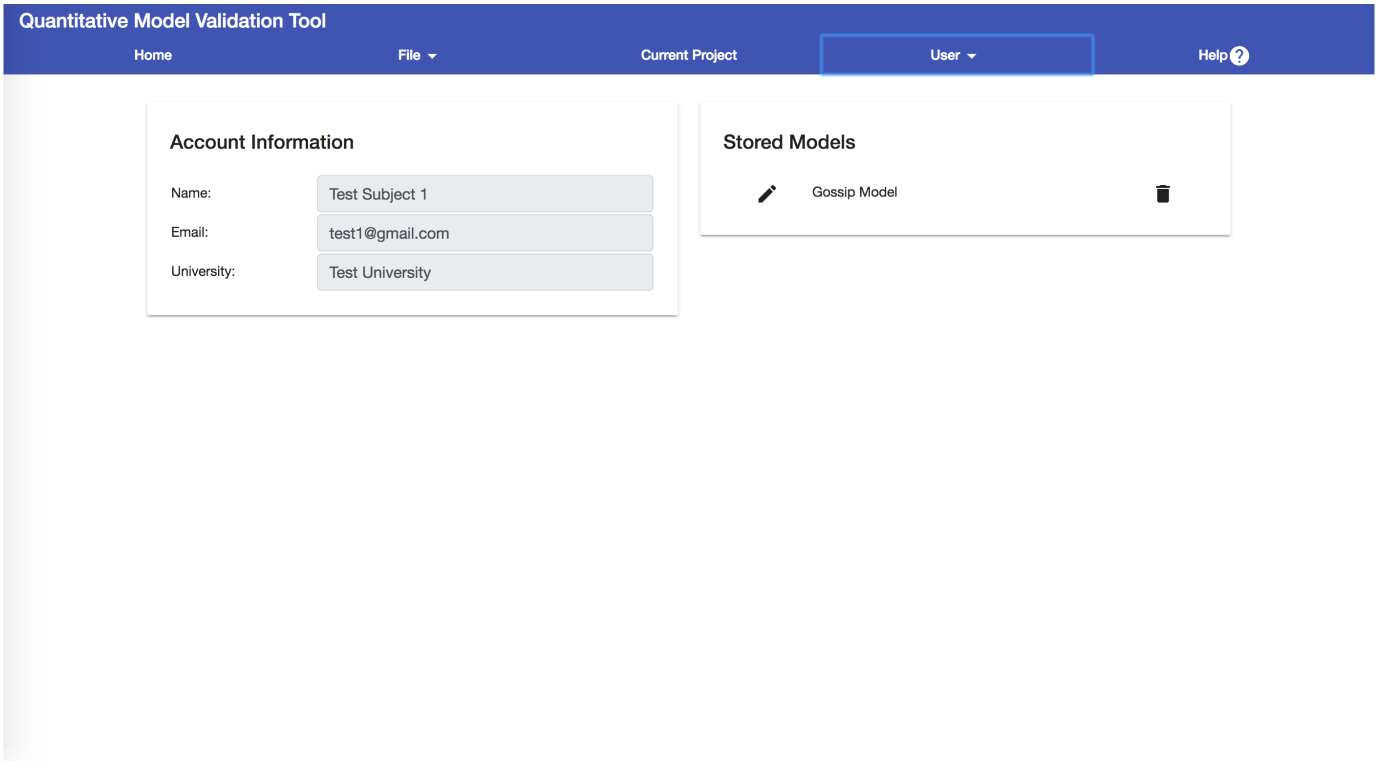

Fig. 13. This is the User Dashboard which displays user data and the user's stored projects.

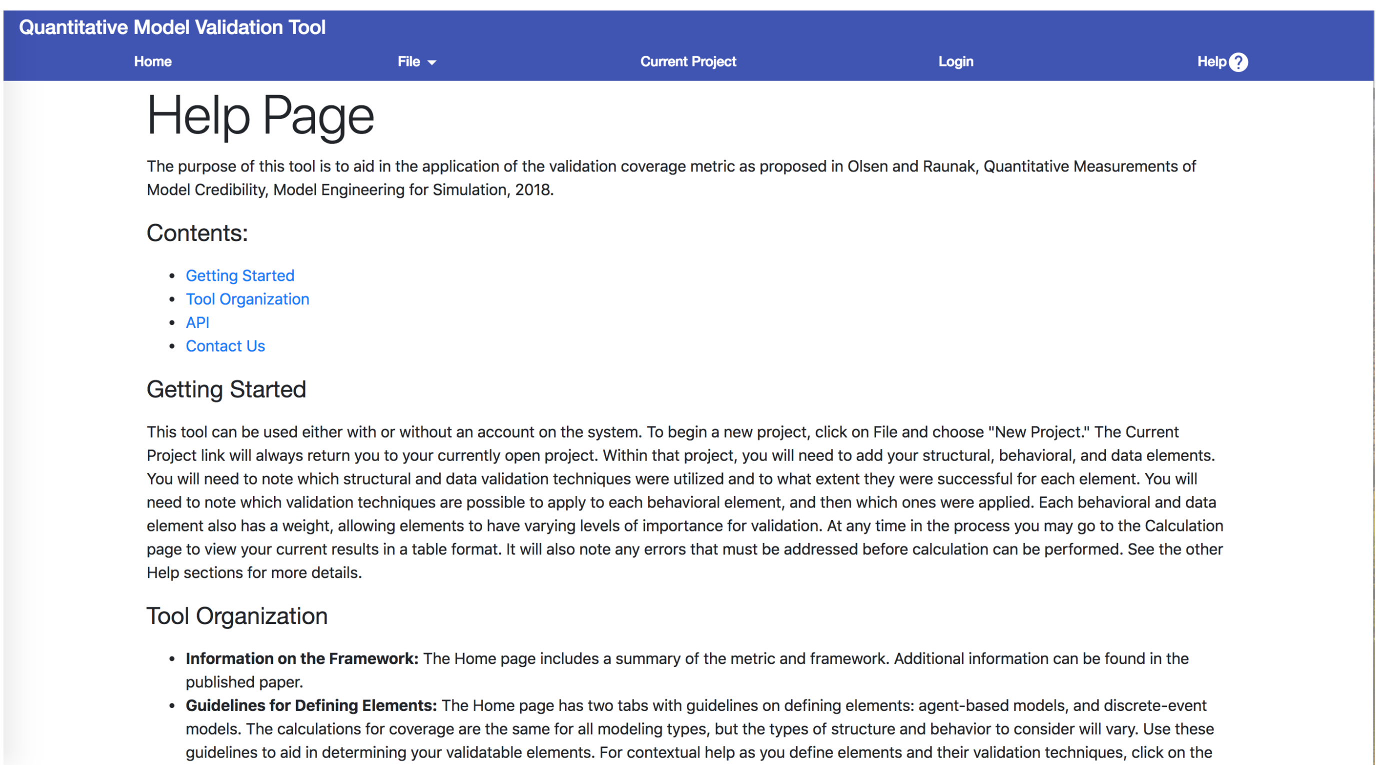

Fig. 14. The Help screen contains information regarding using the online tool and contacting the Loyola Simulation Team. Included in this page is the API's documentation so users can interface with it directly.