Model Comparisons: Docking ORGAHEAD and SimVision

Marcus A. Louie & Kathleen M. Carley
Carnegie Mellon University
mlouie@andrew.cmu.edu, kathleen.carley@cmu.edu

Laleh Haghshenass, John C. Kunz, and Raymond E. Levitt
Stanford University
lalehh@stanford.edu, kunz@stanford.edu, : rel@cive.stanford.edu

Abstract

Comparing computational models is an important method for building scientific knowledge about the design, construction, and modification of computational models. The model comparison process, however, is not well worked through. We illustrate a method of model comparison referred to as model alignment, or 'docking,' using two computational models of organizational design and evaluation, Docking serves three purposes, 1) To illuminate implicit assumptions, 2) to determine the points of similarity and differences in the models, and 3) to determine the ways that the models can be used in a complementary fashion. Results underscore how implementation differences between models can affect their outcome measures.

Contact:
Marcus A. Louie
Department of Engineering and Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213

Tel: 1-412-268-3940
Fax: 1-412-268-3938
Email: mlouie@andrew.cmu.edu

Key Words: Organizational design, simulation, validation, model alignment, docking.

Support: The research reported herein was supported by the Office of Naval Research (ONR) Grant No. N00014-97-1-0037, the National Science Foundation Grant No. KDI IIS-9980109 and IGERT 9972762 for research and training in CASOS. Additional support was provided by the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University (http://www.casos.ece.cmu.edu). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the National Science Foundation or the U.S. government.

Model Comparisons: Docking ORGAHEAD and SimVision

Marcus A. Louie, Kathleen M. Carley, Laleh Haghshenass, John C. Kunz, and Raymond E. Levitt

I. Introduction

Computational models designed to explain similar phenomena may use different representations of the data, process the data differently, and vary in the assumptions they make. While the models are supposed to explain the similar phenomena, these differences may produce inconsistent results in specific parts of the input space. It is important to be able to compare and contrast different computational models in order to map out where these differences lie and what the source of the differences are. This is often thought of as model –to –model analysis. One way to perform a model-to-model analysis is through the alignment of computational models, or what is sometimes referred to in the computational organization literature as *docking* [1]. Docking is a term coined by Axtell et al. to describe the process by which two models are made to give equivalent results.

Establishing equivalence, however, is not the most valuable part of the docking exercise. The process of docking, regardless of equivalence outcome, is perhaps the most fruitful part of the endeavor. The process lays bare the similarities and differences between the two models, uncovers implicit assumptions made in each model, and forces the clarification of semantic differences in data representation, making it easier for others to see how the models relate. Future modelers will have gained insight into the effects of various computational features. In addition, by uncovering the operational and representation differences we can understand the extent to which each affects model outcomes. In this sense, the docking process serves as a sensitivity analysis of model features on model outcomes. Ultimately, improved understanding of the models and their relation to one another facilitates our ability to use the models in complementary ways.

Following the seminal work of Axtell, Axelrod, Epstein, and Cohen who docked the Sugarscape model and the Axelrod Culture Model (ACM), the current paper reports the results of a current drive to dock the SimVision and the ORGAHEAD 2002 models. Establishing equivalence gives the models' designers confidence to say that their model can reproduce the other model's results. This gives both models a greater sense of validity.

The complexity of organizational design has made the use of computational models attractive to theorize about different organizational forms and how they interact with their environment. ORGAHEAD 2002 and SimVision are both computational models of organizational design. While striking differences exist between SimVision and ORGAHEAD 2002, both models can support the same types organizational forms, have a concept of a task, and model individual actors with access to certain resources or skills to solve tasks. How these are represented in the two models differ greatly, and it is these differences that may affect significantly both the equivalence results their interpretation.

To lay the foundation for the paper, we describe different types of docking, give a brief description of ORGAHEAD 2002 and SimVision, and describe the data used to dock. Measurements by which the two models will be compared are explained and finally, the results and observations from the docking process are presented.

II. Background

A. Types of Docking

The docking study performed by Axtel et al. represents but one type of docking investigators can perform. While ACM and Sugarscape have great differences, they share enough similarities to allow the direct comparison of the outputs. When the models do not share the same outputs, the comparison of outputs must between outputs that are expected to relate systematically to each other. In other instances, the purpose of the models may complement each other in a way that allows the outputs of each one to be used as the input into the other. Still another notion of docking models is when two models can be combined into a meta-model. This can happen when two models predict similar phenomena the same way, but in addition, each individually makes separate predictions. The next subsections describe each of the docking types.

i. Comparison. The comparison approach is the basis for what the original docking method employed by Axtel et al. Two computational models are given the same input data, the models are run, and the output is collected. Before analysis of the output occurs, it is checked for the degree of equivalence. Equivalence in output can be checked three different ways: numerically, statistically, and relationally.

Numerical equivalence compares the output and checks to see if they are the same. It only makes sense to check for numerical equivalence between models that do not use stochastic processes, as models using stochastic processes will only give the same result y chance. As most complex models are stochastic in nature, attempting to establish numerical equivalence is rarely performed.

For instance, when stochastic models are being compared, *statistical equivalence* can be used. The models should give outputs that use in the same units. The models will need to be run multiple times in order to generate a distribution of output for each model. A statistical test is then used to check whether the distributions of the outputs are the same.

When the units of the models are not the same, testing for statistical equivalence is not applicable. It is possible to check for a weaker form of equivalence, *relational equivalence.* Relational equivalence is established when the output of the models change in consistent directions as the input changes. For example, two models exhibit relational equivalence if the output of both models is exponential over time.

After equivalence is checked, an analysis should occur that seeks to explain the results of the equivalence test. It is likely necessary that more experiments will need to be run in order to explain when the models give equivalent results, when they do not, and what features of the models are responsible for the similarities and differences.

We use the comparison based method in this study, with the goal of being able to combine the two models into a meta-model.

ii. *Integration*. The integration form of docking combines two or more models in order to model more complex systems than either of the models can do independently. If each model can use as input some output of the other model, then there is basis for integrating the two models. The integration approach to docking is most useful when the data going in and coming out of the models happens continuously in a cycle. For example, referencing Figure 2, the flow of data in an integration form of docking can happen in the following way: $Input_A$ is given to $Model_A$ which processes the data and gives $Output_{A.}$ $Output_A$ is then used as $Input_B$ into $Model_B$. The output of $Model_B$, $Output_B$ is then used as $Input_A$ and then the cycle starts again.

An example of a current integration effort is between SimVision and Blanche. Blanche is a computational model of the co-evolution of networks. It has the ability to model transactive memory in an organization. Currently, SimVision does not model transactive memory, its inclusion is reasonable considering it models team work processes. With some code changes to the models to allow them to share data automatically, SimVision made use of Blanche's transactive memory system.

A very similar approach to integration is known as interoperability. This can be done when the output of one model is used as input into the second, but second's output is not used as input into the former.

iii. *Meta Model*. Another way to combine models is to demonstrate that two models have at least some output in common given the same input and that they both predict the same output (established by a test of equivalence). When the outputs of the models are dependent on all of the input then the two models can be treated as a single meta-model. The meta-model takes as input the same data as the individual models but has the feature of producing both model's output.

B. The Models

i. *ORGAHEAD 2002*. ORGAHEAD 2002 is an organizational learning model. It tests the ability of organizations under different forms to perform a classification task and adapt to their environment. Tasks are presented to an organization one at a time. Each member has access to certain parts of the task and reports its opinion on the true state of the task to their superiors. Organizations are capable of adapting: organizations may change their structure over time, and actors have the ability to learn and change their behavior over time. ORGAHEAD 2002 is well suited for theorizing about organizations by examining underlying dynamics.

For more details on the ORGAHEAD model see [3], [4], and [5].

ii. *SimVision.* SimVision models organizations, specifically project-based teams, where the tasks are relatively routine. Activity interdependence is analyzed to see how coordination requirements change and organizational design and communication methods analyzed to see how they can alter coordination capacity. Actions and interactions between actors are simulated as functions of attention allocation, actor capabilities, and communication. The performance of the organization is measured by how long it takes the organization to complete its tasks, the cost of the task, and quality of the communication while solving the task.

For more details on SimVision, also referred to as the Virtual Design Team (VDT), see [2] and [6].

III. Data

Data were gathered from the Naval Post-Graduate School (NPS) on teams performing simulated take-the-beach exercises. Three organizational forms were modeled from the teams participating in these exercises. The forms differed in their authority structure (who reports to whom), communication structure (who talks to whom), capabilities structure (what skills people have), and task structure (who is assigned what task). To compare each structure, the task environments were constant across the three teams. The tasks environment was described by the partial ordering in which the tasks must be completed and the skills needed to complete the tasks.

For identification purposes, the teams were labeled A06, A14, and A16.

IV. Virtual Experiment

A. The Models

Each of the teams, A14, A06, and A16 were constructed in ORGAHEAD 2002 and SimVision according to the authority structure, capabilities structure, and task structure given by the data. Most of the encoding into the models was straightforward. At some points during the encoding, however, decisions had to be made about how the data should be represented in the model.

The most important decisions that were made dealt with who were assigned to the tasks in SimVision. SimVision makes important requirements about the nature of task assignments. First, it assumes that if a task exists then somebody is assigned to it. In the data, all 29 tasks are assigned to at least one actor when the three teams are considered together. For individual teams, however, not all 29 tasks are assigned to at least one actor, forcing the decision of who should be assigned those tasks in SimVision. For these experiments, the actor with the best skill match to a previously unassigned task was assigned to it. An alternate reasonable task assignment could have given the task to the actor with the lowest cognitive load.

The second assumption is that tasks are assigned to only one actor. For several of the tasks in the data, tasks are assigned to multiple actors. Again, the decision of who is assigned the task in SimVision needed to be made. Arbitrarily, for each team the task was assigned to the first actor that appeared in the data who was assigned to the task. The remaining people assigned to the task in the data were given secondary task assignments to the task. Actors will only work on secondary tasks when they do not have any work to do on their primary task assignment. In general, all task assignments are considered to be primary task assignments unless specified as being secondary.

Decisions were also made with respect to how to encode skill requirements. In the data, a task can have one or more skills associated with it. If an actor has a skill that a task requires, then the actor will perform the task with less probability of an error. Tasks in SimVision have a single principle skill associated with them as opposed to the sometimes several skills associated with a task given in the data. The choice then was which skill should be associated with the task. Given no grounds for choosing, the first skill that appeared in the skill requirements meta-matrix for a given skill was selected.

Normally, virtual experiments in ORGAHEAD 2002 are designed to allow organizations to make changes. In our experiments we prevented organizations from making changes in order for the results to be more meaningful when compared with those from SimVision.

B. Measurement

One of the challenges of comparing these particular models is the difference in measurements that the models produce. SimVision produces an estimated actor backlog, a measure of how much work an actor has been assigned to do but has not started yet. In ORGAHEAD 2002, actors work on a one task at a time so there is no concept of backlog. The differences in what the models measure prevents us from checking for “numerical identity” or a “distributional equivalence [1], however we can still perform a relational equivalence provided there are measurements from the two models that lend themselves to being compared.

One of the primary measures that SimVision outputs is the estimated duration of a project. How long a project takes to complete depends on a number of factors, including the amount of rework that needs to occur because of failed subtasks, the amount of coordination needed between actors (which itself is contingent on other aspects), level of interdependence between subtasks, and the difficulty of the subtasks. While ORGAHEAD 2002 does not explicitly consider duration of tasks, it does consider how accurate an organization is at classifying the tasks. Actors in ORGAHEAD 2002 have the capability to learn, thus we expect that given enough time, an actor will be able to eventually learn to classify those tasks it got wrong. They may still misclassify due to attention and memory limitations, but their classification rate will approach a maximum ceiling. How many tasks an actor and organization must relearn to classify depends on how many tasks they classified correctly in the first place. If we assume tasks take a constant amount of time to classify, then we can assume that organizations that have higher classification accuracies also require a shorter amount of time to relearn the tasks they misclassified. Thus, a reasonable relationship exists between classification accuracy and task duration.

The ORGAHEAD 2002 notion of accuracy and the SimVision notion of duration are defined below:

Accuracy(t): The percentage of all tasks seen that the organization or actor correctly classifies, where *t* is the number of tasks seen so far.

Accuracy(*t*) = 100 * (Number of all tasks correctly classified / *t*)

Duration: The estimated time, measured in days, that the organization is expected to complete all of its tasks.

V. Results

Our intuition that classification accuracy in ORGAHEAD 2002 is related to task duration in SimVision is supported in this set of experiments. We note that while the models agree using this data set, it is quite reasonable to suppose that the models might disagree on a different data set.

While we managed to demonstrate relational equivalence between ORGAHEAD 2002 and SimVision, the experiments we constructed were based on an encoding of the data that could have been done differently. Several encoding decisions were arbitrary so another encoding, if used, could have altered the results.

In this section, we check to see how our decisions to handle multiple actors being assigned to work on the same task may have affected the results. SimVision is designed to handle one person per task, so a decision was made to assign one person, arbitrarily, as being the primary person working on the task. The remaining actors assigned to the task in the data were assigned as secondary actors to the task. The data did not have a notion of primary and secondary task assignments. Furthermore, in ORGAHEAD 2002 tasks can have multiple actors assigned to them, so the use of secondary actors in SimVisoin to emulate a task assignment that does not distinguish between actors' access to the task was investigated.

A set of virtual experiments was designed to gain insight into the use of secondary actors for multiple actors assigned to a task. The virtual experiments were based off of modifications to team A14 in task assignments. Virtual experiments were run in both SimVision and ORGAHEAD 2002.

The virtual experiments demonstrated that the effect of acting as a secondary actor on a task depends on whether the actor will already have work to do while their secondary task assignments are active. Part of the effect we believed was due to the nature of the tasks being worked on. If instead the tasks receiving secondary actors were non-critical, in the sense that how quickly the task is completed affects minimally when other tasks begin or how long they take to complete, we should expect a minimal at most difference in results. Additional experiments were designed in SimVision that changed the tasks receiving secondary actors to non-critical tasks. Unexpectedly, the results were exactly the same as when critical tasks were selected. The result was driven by the nature of the tasks that the primary actor was working on. While the secondary actor was assigned to non-critical tasks, the actor who the secondary actor was helping also had critical tasks to work on. The secondary actor was able to relieve more time for the primary actor to work on their critical tasks.

These results suggest that which actor receives the primary task assignment when multiple actors are assigned to a task can affect the results. In other words, arbitrary decisions about how to deal with multiple people being assigned to a task are important decisions. The selection of the primary actor that results in the other secondary actors not having time to work on their secondary task assignments will lead to different results than the selection of a primary actor that leaves the secondary actors free to work on the task.

References

[1] R. A. Robert Axtell, Joshua M. Epstein, Michael D. Cohon, "Aligning Simulation Models: A Case Study and Results," *Computational and Mathematical Organization Theory*, vol. 1, pp. 123-141, 1996.

[2] R. E. L. John Kunz, Y. Jin, "The Virtual Design Team: A Computational Model of Project Organizations," *Communications of the Association for Computing Machinery*, vol. 4, pp. 84-92, 1998.

[3] K. M. Carley & D. M. Svoboda, "Modeling Organizational Adaptation as a Simulated Annealing Process." *Sociological Methods and Research*, 25(1): 138-168, 1996

[4] J. Lee & K. M. Carley, 1998, _Adaptive Strategies for Improving C2 Performance_ Pp. 66-77 in Proceedings of the 1998 International Symposium on Command and Control Research and Technology. Conference held in Monterray, CA, June, 1988. Evidence Based Research, Vienna, VA

[5] K. M. Carley & Y. Ren, 2001, "Tradeoffs Between Performance and Adaptability for C3I Architectures." *Proceedings of the 2001 Command and Control Research and Technology Symposium*. Conference held in Annapolis, Maryland, June, 2001. Evidence Based Research, Vienna, VA.

[6] R.E Levitt, G.P. Cohen, J.C. Kunz, C.I. Nass, T. Christiansen, and Y. Jin, "The Virtual Design Team: Simulating How Organization Structure and Information Processing Tools Affect Team Performance", *Carley, Kathleen M. and M.J. Prietula, eds., Computational Organization Theory*, pp. 1-18, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.