A Docking Study of SimVision and ORGAHEAD

DRAFT

Marcus A. Louie
17-752: Dymanic Organizations
17 December 2002

1 Introduction

Following the efforts of Axtell, Axelrod, Epstein, and Cohen (1996) to “dock” 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 models. 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 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 process of docking, regardless of equivalence outcome, is perhaps the most fruitful part of the endeavor. The docking process lays bare the similarities and differences between the two models, 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.

While striking differences exist between SimVision and ORGAHEAD, 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, but that is part of the reason why attempting to dock these two models can be so valuable.

To lay the foundation for the paper, a brief description of the two models is given first, followed by a description of 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.

Citation: Marcus A. Louie, 2002, A Docking Study of SimVision and ORGAHEAD

2 Background

2.1 Types of Docking

The docking study performed by Axtel et al. represents but one type of docking that investigators can perform. While the Axtel Cultural Model 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 label and describe each of the docking types.

2.1.1 Comparison

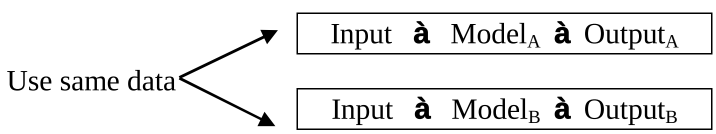


Figure 1: Docking by comparison

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 stoachastic 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 instances 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.

If the units of the models are not the same, testing for statistical equivalence does not makes sense. It is still possible to check for relational equivalence, that is, do the output of the models change in consistent directions when 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.

2.1.2 Integration

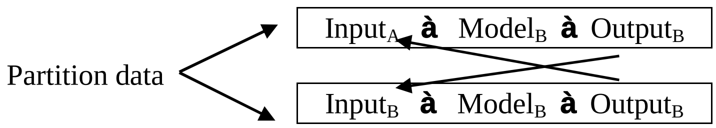


Figure 2: Docking by integration

The integration form of docking allows models to be combined to potentially model more complex systems than either of the models individually. 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.

2.1.3 Meta-Model

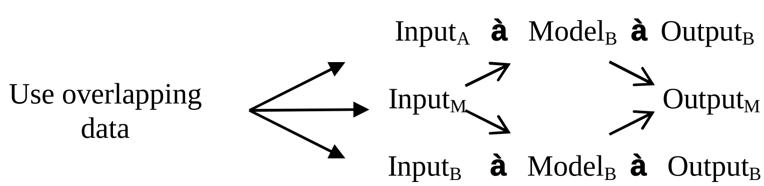


Figure 3: Docking by creation of a meta-model

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.

2.2 The Models

Our aim is to determine the extent to which it is possible for the SimVision model built by Raymond Levitt et al. and the ORGAHEAD model built by Carley et. al to explain the same phenomena given the same input. The two models differ in important ways. For example, ORGAHEAD supports the ability for organizations to alter their structure periodically through a variety of means. These means include the ability to hire and fire personnel, changing what resources are available to whom, and changing which people are assigned to what tasks. Organizations in SimVision, on the other hand, do not change their structure for the lifetime of the organization.

The exposition of the two models given here is intended for two types of readers, those who are familiar with the models but need a review and for those who have no prior knowledge the models. In both cases, the background information is intended to be sufficient to understand the challenges to docking these two models. See prior papers on ORGAHEAD (Carley 1996, Carley & Svboda 1996) and SimVision (Jin & Levitt 1996) for a more comprehensive description on the respective models.

In both expositions, the task environment, operational level, and strategic level of the models are described.

2.2.1 Virtual Design Team

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.

<u>Task Environment</u>

Tasks are composed of multiple a*ctivities* in SimVision. For example, a task in real life might be the design of a refinery (Levitt et al. 1994). The activities that go into the design might be the chemical process design, piping design, and structural design. Each of these processes requires some amount of information processing and may or may not require the coordination of multiple people.

Activities are described along a variety of dimensions. The dimensions of an activity are best described by Kunz et. al (1998):

> The SimVision activity model represents (parentheses show type of attribute value):
> - Duration (nominal time)
> - Failure dependence (list of activities)
> - Requirement complexity (low, medium, high)
> - Required skill (e.g., financial accounting, structural steel design)
> - Solution complexity (low, medium, high)
> - Subtask size (time to do one subtask within the activity, where activities are assumed to decompose into equal sized subtasks, and a subtask is the minimum amount of work that can be determined to have "failed.")
> - Successors/Predecessors (list of activities)
> - Uncertainty (low, medium, high)
> - Work volume (time for an actor with "medium" level of "required skill" to perform the activity assuming no rework)

Coordination amongst individuals is also explicitly modeled. The complexity of an activity results in a certain probability that a subtask will fail. Subtask failure results in communication about the failure to the responsible person's supervisors. If a decision is made to rework the subtask, then rework is propagated to the "failure dependent" activities. Activities also have a degree of uncertainty and interdependence associated with them, which may lead to varying levels of communication with others.

Operational Level

The structure of an organization is defined by the authority structure (who reports to whom), communication structure (who talks to whom), resource or skill structure (what resources does each person have), and a task structure (who is assigned to what tasks).

A simulator in SimVision creates “work item”-communications that notify actors when a subtask of an activity they are responsible for is ready to be worked on. Actors may have a number of work items to complete, so they choose one according to an attention rule and may initiate communication with others according to the communication structure. When a subtask fails, the responsible actor reports the event to a supervisor according to the authority structure. How likely a subtask fails will depend on the actor’s skills and the complexity of the subtask. The organization is finished when all activities are finished.

Activity subtasks and communications for an actor are held in their “in-tray” until they are ready to be processed. The attention rules of an actor determine whether to interrupt the current activity upon the arrival of a new item and also determine which item to process next after completing their current item. What items get attention is a factor of the priority of the current activity, priority of the incoming item, and the order that items arrived in the in-tray. The attention rules give actors a boundedly rational behavior when choosing what to work on next.

Strategic Level

Organization and actors In SimVision are not strategic in the sense that they attempt to modify their behavior in order to improve their performance. The various structures that define an organization and its actors remain constant throughout the span of its life.

2.2.2 ORGAHEAD

ORGAHEAD 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 is well suited for theorizing about organizations by examining underlying dynamics.

Task Environment

Organizations are presented with classification tasks. The tasks are represented as a bit array of zeros and ones. Classifications occur according to predefined decision rule that accounts for the composition of ones and zeros in the task. An example decision rule would give an answer of 1 if there were more ones and zeros, and 0 otherwise. This type of rule is called a “majority rule”. This decision rule did not factor in the position of the bits, but other decisions rules could be designed to account for bit positions.

For illustrative purposes, a common mapping of the ORGAHEAD classification to a real-world situation is the radar classification task. In the radar classification task, the organization must decide whether a given stimulus is friendly or hostile based on a series of sensors. Different individuals may be responsible for monitoring different sensors, and these individuals pass their results on to one or more superiors. The results may be the speed or direction of an aircraft. In ORGAHEAD, these numbers are reduced to bits of zeros and ones, which might represent answers to questions such as, "Is the aircraft's speed greater than 400 knots?"

Operational Level

ORGAHEAD

The structure of the organization is defined by an authority structure and a resource structure. According to the resource structure, actors receive information about a task. For example, the resource structure might dictate that an actor has access to the second, third, and fifth bit of a task. Actors then pass their decision of the true state of the task up to their superiors in compliance with the authority structure. The decision of the true state on behalf of an actor is based on the predefined decision rule.

The communication structure that exists in SimVision is not present in ORGAHEAD, rather the communication structure and the authority structure are the same. Authority structures can follow a wide range of forms, but in the docking experiments at hand, only hierarchical structures are used.

Each actor of the organization is modeled as a boundedly rational being, with limits on their attention (how many task bits and resources they can consider) and memory (how many tasks' outcomes they remember).

ORGAHEAD 2002

Whereas in ORGAHEAD, every actor of an organizations works on tasks together, ORGAHEAD 2002 supports the notion of individual actors being assigned to specific tasks, such that not every actor will work on every task. For example, an actor assigned to tasks A and B will only work on those two tasks.

The authority and resource structures operate the same in ORGAHEAD 2002 as in ORGAHEAD, and actors are made boundedly rational in the same ways.

Strategic Level

Given a particular task, the performance of an organization is dependent on its design. The design of an organization can be described by a number of different features, such as its size, number of levels in the authority structure, average number of subordinates per supervisor, and average number of resources assigned to a person. The features of the organization form a multidimensional performance surface that may include a number of peaks and valleys. As its goal, the organization seeks the maximum peak, which represents a set of values for the given features.

In ORGAHEAD, organizations strategically change their design, traversing the performance surface for a higher position. Changes are proposed periodically and evaluated. The evaluation consists of looking ahead by hypothesizing about the future given the organization and the proposed change. If the expectation is that the proposed change will benefit the organization, then the change is implemented.

As time passes, the organization is less likely to accept proposed changes. A simulated annealing process controls the likelihood that an organization will accept the proposed changes, whereby initially an organization is most likely to accept proposed changes and then gradually becomes more risk adverse. The increased risk aversion is manifested in a decreased likelihood of accepting proposed changes.

2.3 Key Differences – SimVision and ORGAHEAD 2002

Several adjustments to ORGAHEAD were made to come as close to simulating the conditions that organizations in SimVision experienced. The points of difference that most seriously needed to be addressed were the issues of task selection, task precedence, task assignment and organizational adaptation.

Tasks in ORGAHEAD are considered to be events that occur exogenously to the organization that the organization must make some decision about. These tasks, or events, can be thought of as manifestation of the environment that cannot be predicted, and thus their order is not known in advance. Tasks are therefore randomly selected with replacement from the set of possible tasks. In SimVision, the set of tasks that the organization must complete is known in advance. The result is that the three organizations participating in the virtual experiment will see the same tasks. This is likely not to be the case in ORGAHEAD. We addressed this issue by predetermining the tasks seen by the organization.

In SimVision, tasks follow a certain precedence ordering. The ordering can be thought of as a partial ordering, where some tasks can be worked on concurrently while other tasks require the completion of one or more other tasks before starting. In ORGAHEAD, because the tasks were originally randomly selected, no task required the completion of another task in order to be completed itself.

The presentation order of tasks in ORGAHEAD was carefully chosen to reflect the task precedence requirements. Thus, tasks appeared in the task list only after all the tasks that were required to finish first appeared. Tasks that did not depend on each other were placed randomly in the list subject to the above condition.

Tasks are assigned differently between SimVision and ORGAHEAD. In SimVision, actors can be assigned to one or more tasks and may not be assigned to any. Actors in ORGAHEAD, on the other hand, work on each task together. In other words, actors in ORGAHEAD are not assigned to specific tasks to work on. Rather they work on all tasks. To understand the difference this makes, ORGAHEAD.V2 supports individual task assignments, and actors will only work on a task that they have been assigned to.

The two models also differ in how explicitly the task environment is modeled. SimVision models the task environment much more explicitly than ORGAHEAD, permitting tasks to require a certain set of skills in order to work on them. Subtask failure in SimVision results in coordination with other actors, affecting the other actors' work as well. While actors in ORGAHEAD classify a task together, individual mistakes or inabilities to classify the task only affect the group as a whole, not the performance of other individuals. This inability to affect may be a problem in docking the two models at the actor level.

Some points of difference were not possible to address without significant modification to the underlying theory of ORGAHEAD. For example, in SimVision team members can communicate laterally with each other to complete tasks. In ORGAHEAD, team members only communicate through the authority structure, where subordinates report the result of their classification task to their superiors.

Tables 1 – 4 summarize the similarities and difference between the two models.

Table 1: Actor Features

	ORGAHEAD	SimVision
Actors	Yes	Yes
Actors have skills	Knowledge 0/1	Knowledge scale [0-1]
Actors learn new skills	Yes	No
Authority structure	Yes	Yes
Actor responsibility	[0-n]	[1] Assume first specified is primary; others secondary
Actor gets feedback	Yes	No
Actors gain experience	No	Yes (not used in experiment)
Actors have roles	Yes	Yes
Order of processing	FIFO	Mixed
Actors can forget	Primacy, recency	

Table 2: Task Features

	ORGAHEAD	SimVision
Tasks	Yes	Yes
Variable task duration	No	Yes
Task required skills	[1-n]	[1] Assume first required
Task precedence	Yes	Yes
Task accuracy	Yes	No
Actors per task	[1-n]	[1]
Task rework	No	Yes (not used in experiment)

Table 3: Operational Features

	ORGAHEAD	SimVision
Authority structure	Yes	Yes
Communication structure	Yes	Yes
Authority = communicate	Yes	No
Actor responsibility	[0-n]	[1] Assume first specified is primary; others secondary
Lateral communication	No	Yes
Task requirements	Changed by environment (pre-defined in experiment)	Pre-defined

Table 4: Strategic Features

	ORGAHEAD	SimVision
Hire/Fire actors	Yes	No
Task reassignment	Yes	No
Resource reassignment	Yes	No

3 Data

Three organizational forms were modeled. The forms differed in their authority structure, communication structure, capabilities structure (what skills people have), and task structure (who is assigned what task). To compare the merits of 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. The authority structures are given in Figure 4. A more detailed picture showing the authority structure, task assignment, and task precedence ordering for team A06 is graphically illustrated in Figure 5.

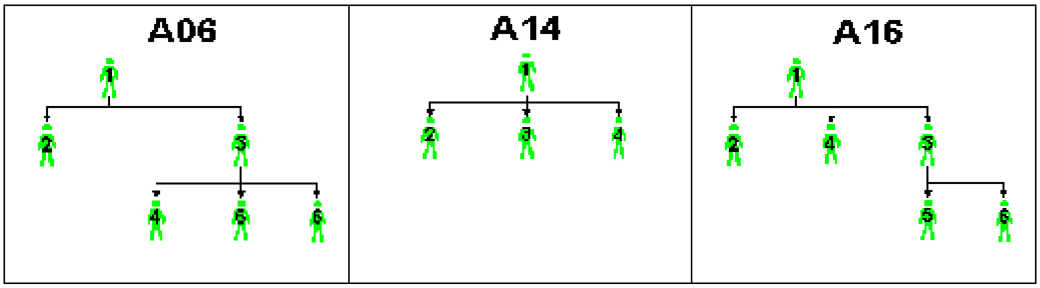


Figure 4: Authority structures for A06, A14, and A16.

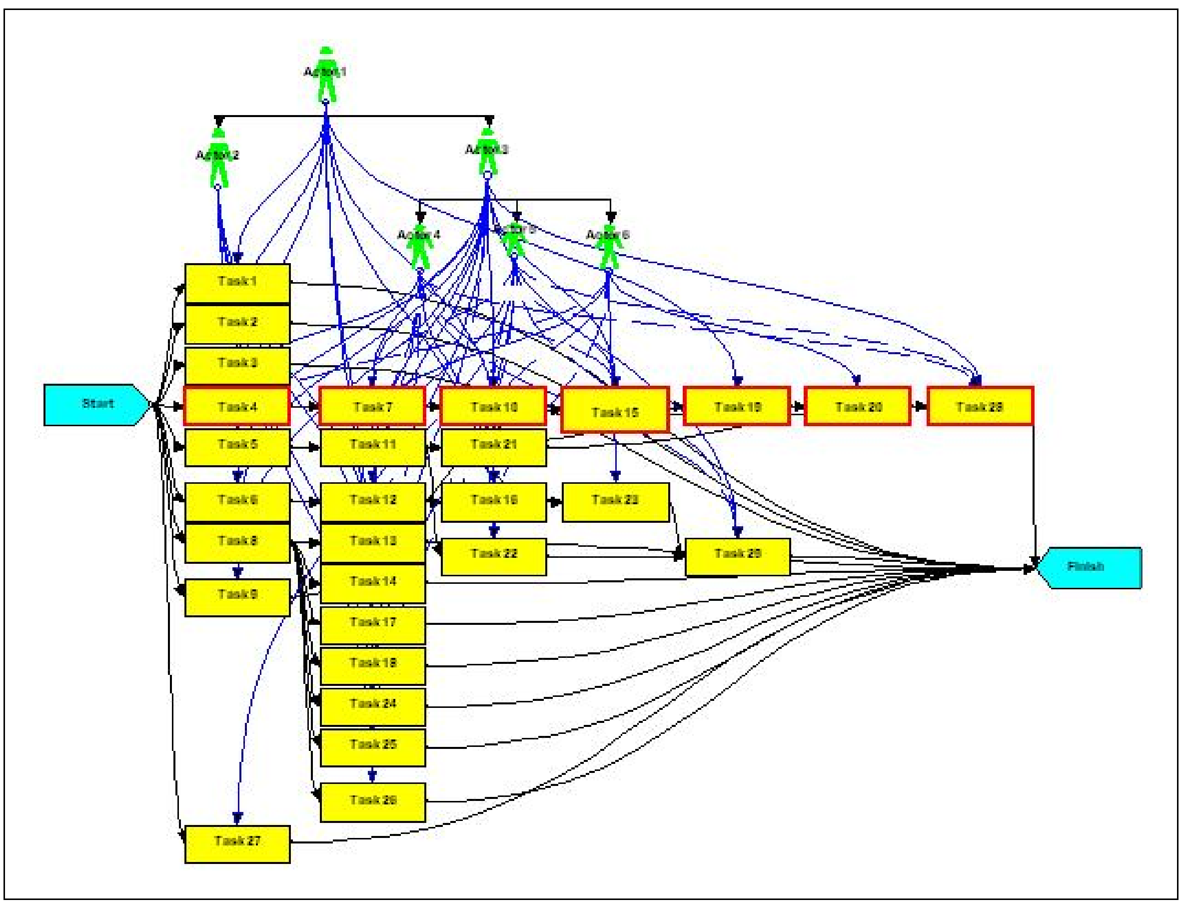


Figure 5: A06 - Authority, task assignment and task precedence structure.

4 Virtual Experiment

<u>SimVision</u>

Each of the teams, A14, A06, and A16 were constructed in SimVision according to the authority structure, capabilities structure, and task structure given by the data. Most of the encoding into the model 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. 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 single 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. This decision point does not arise in ORGAHEAD as it allows multiple people to be assigned to a task. 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. Figure 6 shows the task assignment meta-matrix for team A06. Columns 4, 19, and 24 show tasks that are not assigned to anyone. The yellow highlighted cells mark who in SimVision were assigned those tasks. Columns 7, 10, 12, 14, 15, 25, 28, and 29 show tasks that were to assigned to multiple people. The first cell with a "1" in those columns marks the associated actor given the primary task assignment. The blue cells mark those people given secondary task assignments.

Yellow cells show tasks assigned by default to the actor with the best skill match.
Blue cells show tasks and their assigned secondary actors
Task actor assignments A06 (actor x task)

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
3	0	0	0	0	0	1	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1
4	0	0	0	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1
5	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1
6	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

Figure 6: Deciding who are assigned what tasks in SimVision.

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.

SimVision supports giving probabilities to certain events occurring. These events are central to the operations of the model and are not specified by the data. The information exchange probability measures the level of communication in the project between positions that are responsible for tasks linked by communications links. The information exchange probability was set to 0.5. Noise probability measures the likelihood that work on a project is interrupted by events that take time away from work related tasks. The noise probability was set to 1.0. Functional error probability is the probability that a task will be fail and need rework. The functional error probability was set to 0.1.

Figure 5 shows a screenshot of team A06 coded into SimVision.

ORGAHEAD

ORGAHEAD was developed to allow organizations to adjust their structure to the demands of the environment. Adjustments organizations can make include changing who reports to whom, what resources people have, what tasks people are assigned to, and hiring and firing people. For these experiments, we disallowed organizations to make any adjustments to their structure in order to allow for more meaningful comparisons with the actors in SimVision.

Relative efficiency reports were given after every block of 290 tasks. We refer to each block of 290 tasks as a change cycle, since normally after the efficiency reports organizations are given an opportunity to change. Organizations worked through 80 change cycles, for a total 23,200 tasks.

Each block of 290 tasks consisted of 29 distinct tasks, each task appearing 10 times. Each task was 26 bits wide, where each bit represented a resource. A “0” represented that the resource was not needed to complete the task. A “1” represented that the resource was needed to complete the task. Tasks were repeated 10 times to allow actors to learn.

Actors in ORGAHEAD were modeled to be boundedly rational, with limits on their memory. Actors were only able to remember the correct classification of the last 29 tasks they saw. The memory limit was set somewhat arbitrarily, low enough to prevent actors from having near perfect classification which would entail ceiling effects but high enough to allow the actors to learn.

The correct classification of tasks was also set arbitrarily. Tasks can be classified as either friendly or hostile. A task in this experiment was hostile if it required more than half of the resources and was friendly if it required less than or equal to half the resources. All tasks needed at least one resource to be completed.

For both experimental conditions, ORGAHEAD was run using organizational structures defined by A06, A14, and A16. Each organizational structure was simulated 30 times for each organizational form in each experimental condition.

ORGAHEAD.2002

ORGAHEAD is designed to be flexible in the representation of tasks and resources, in some ways. There are many ways to represent a task. Two of these seem critical vis-a-vis docking with SimVision – 1) each task is represented as a bit or 2) each task is represented as a string of bits. Representing tasks as bits allows us to assign individual people to zero or more tasks.

However, this representation prohibits using a precedence ordering because in ORGAHEAD there is no way to specify a precedence ordering of bits to be worked on.

Representing tasks as a string of bits allows a precedence ordering to be followed, because organizations in ORGAHEAD can be fed strings of bits in any order; i.e., while there is no precedence ordering across bits in a string there is a precedence ordering across strings (however, it is strictly linear). The bits in the task can represent the resources needed to classify the task. For example, a 1 in 5th position specifies that the 5th resource is needed to classify the task. By using this representation, we lose the ability to assign individual people to tasks because we can not constantly modify who is allowed to work on a task. Currently in ORGAHEAD, people are given access to certain bits of tasks and this is determined up front in the parameters and does not change when annealing is not used.

In the above experiments, the second representation was used to allow a precedence ordering. This, however, came at the expense of ignoring the task assignment matrix altogether. The capabilities matrix (people x resources) was used to specify which bits each person had access to. We ran this choice first as it best matched the intent of the way the real groups worked.

We intend to run the second way also and see how much difference it makes in the results.

4.1 Measurements

One of the challenges of comparing these particular models is the difference in measurements that the models produce. For example, 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, 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” (Axtell, Axelrod, Epstein & Cohen, 1996), 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 does not explicitly consider duration of tasks, it does consider how accurate an organization is at classifying the tasks. Actors in ORGAHEAD 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 notion of accuracy is defined below

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

$$\text{Accuracy}(t) = 100 * (\text{Number of all tasks correctly classified} / t)$$

<u>Relative accuracy</u>: The percentage of tasks seen within the last *F* that the organization or actor correctly classified.

Relative Accuracy(t) = 100 * (Number of the last *F* tasks correctly classified / *F*)

Per-person relational equivalence is attempted using the per-person relative accuracies and the resource load of the individuals. Each actor an ORGAHEAD can be thought of as having a ***resource load*** that is a function of the number resources they have and the number of people reporting to them. Up to the cognitive limits of an actor, actors with greater resource loads may be more accurate and may thus equivalently incur less of a backlog in SimVision. The number of resources an actor has in ORGAHEAD is equal to the number of task bits that the actor has access to. Actors also receive information about the task from subordinates.

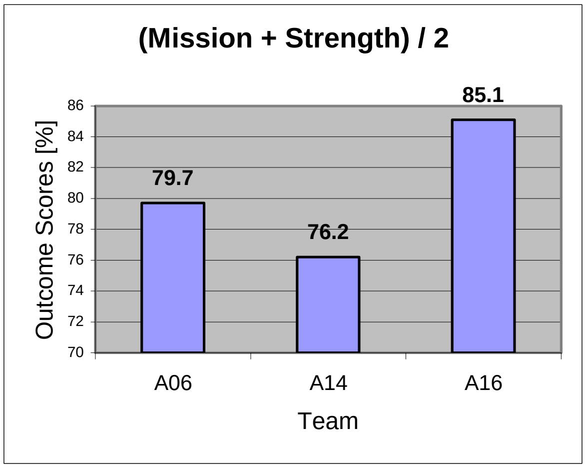
Table 5: Organization Level Measurements

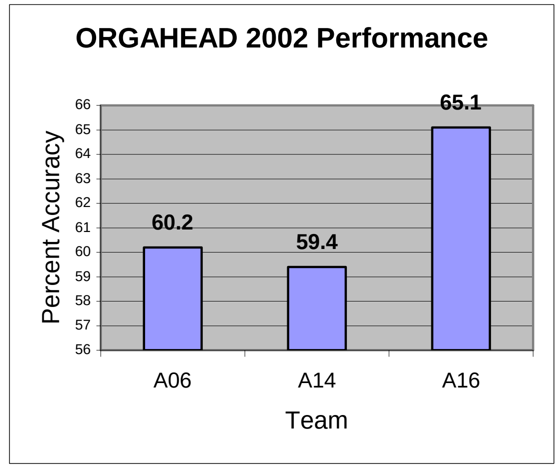
	ORGAHEAD	**SimVision**
Accuracy	Yes	No
Relative accuracy	Yes	No
Workload	Yes	Yes
Rework	No	Yes
Actor backlog	No	Yes
Duration	Partial	Yes
Change in meta-matrix	Yes	No
Measures on meta-matrix	Yes	Yes

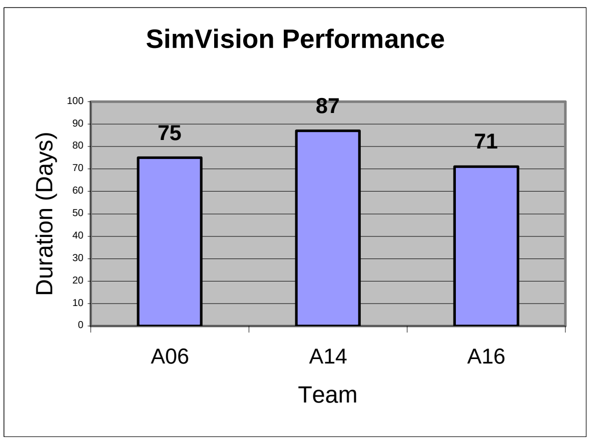
Table 6: Individual Level Measurements

	ORGAHEAD	**SimVision**
Accuracy	Yes	No
Relative accuracy	Yes	No
Actor backlog	No	Yes
Measures of meta-matrix	Yes	Yes

5 Results







SimVision	A14 > A06 > A16
ORGAHEAD	A14 ≤ A06 < A16
Real Data	A14 < A06 < A16

Figure 7: Results of virtual experiments show relational equivalence among SimVision, ORGAHEAD and the real data.

Our intuition that classification accuracy in ORGAHEAD is related to task duration in SimVision is supported in this set of experiments. Figure 7 shows results from the real-world task simulation in the upper-left, ORGAHEAD results in the upper-right, and SimVision results in the lower-left. On the x-axis are each of the teams. Associated with each team is a bar showing the performance as measured in the experiment. While graphically the results from ORGAHEAD and SimVision do not match, recall that higher accuracy ORGAHEAD should relate to lower duration in SimVision. The box in the lower-right summarizes the relations among the results for the different teams. The difference in performance between A06 and A14 in ORGAHEAD is marginal and not statistically significant. Increasing the number of virtual experiments for those conditions may help obtain significance.

6 Analysis

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

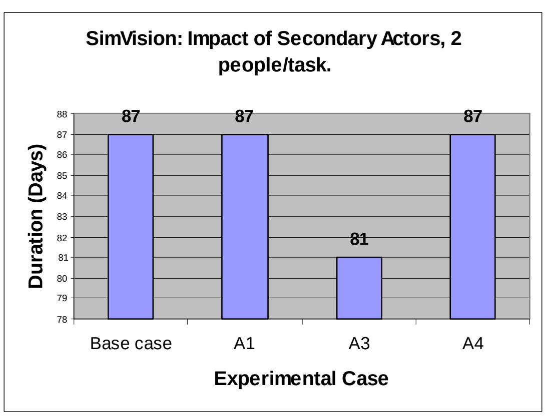
In this section, we check to see how our decisions to handle multiple actors being assigned to work on the same task may have affected the results. SimVision is designed to handle one person per task, so a decision was made to assign one person, arbitrarily, as being the primary person working on the 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 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.

As a base case, A14 was run with only one person per task. The experiment was designed in SimVision by removing all secondary task assignments. The corresponding task assignment changes were made in ORGAHEAD. Next, a set of tasks were chosen such that 1) at some point during a simulation in SimVision an actor existed who was available to work on the set of tasks as a secondary actor, 2) for another actor, there was no point in time during the simulation in which they could work on the tasks as a secondary actor, and 3) each task in the set was "critical" in the sense that a delay in its completion directly delayed the entire project or delayed a number of other tasks that were dependent on its completion. The set of tasks selected were tasks 4, 6, 8, and 9. Their positions in the task precedence structure can be viewed in Figure 5. The primary actor responsible for the tasks was Actor 2.

Three additional cases were developed. Each case assigned one of the other three actors on the team as a secondary actor to tasks 4, 6, 8, and 9. Actor 3 was the actor who had time to spend working on the selected tasks as a secondary actor. Actors 1 and 4 were assigned tasks at times that overlapped when we would expect the selected tasks to be worked on, thus if they were to work on the tasks at all, their contribution would be small. The three additional cases were also coded into ORGAHEAD experiments.

The results of the experiments are displayed in Figure 8. The base case is clearly labeled in the graphs. The labels "A1," "A3", and "A4" denote the cases when Actors 1, 3, and 4 were assigned the selected tasks as secondary task assignments respectively.



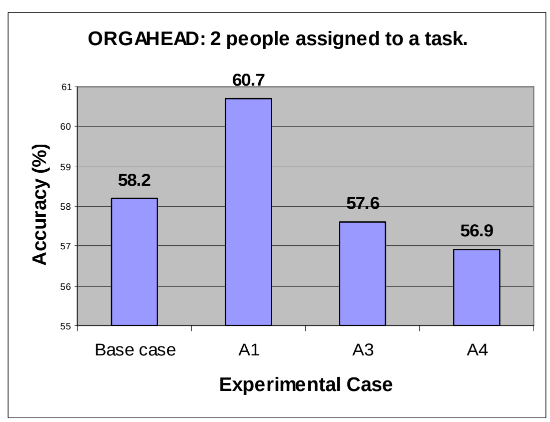


Figure 8: The impact of secondary actors. SimVision uses secondary actors to simulate multiple people assigned to a task. In ORGAHEAD, actors assigned to the same task act have equal priority for working on the task.

As expected in SimVision, the use of Actor 3 as a secondary actor is the only case that produces a difference in performance with respect to the base case. When tasks 4, 6, 8, and 9 are being worked on by Actor 2, Actors 1 and 4 are both working on their primary task assignments so they cannot afford much at all to work on the tasks. Actor 3, on the other hand has free time and does work as a secondary actor on the tasks, reducing the time to complete those tasks. The overall project duration is decreased because the tasks were "critical" to the project.

The previous experiment 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 assignment are active. Part of the effect we believed was due to the nature of the tasks being worked on. If instead the tasks that received 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. The tasks chosen for the set were tasks 1,2,3, and 25. Unexpectedly, the results were exactly the same as when critical tasks were selected – Actor 3 again was able to have an impact on the overall duration. The results make sense when one considers that Actor 2 was also assigned to these non-critical tasks. Actor 3 freed up more time for Actor 2 to work on the critical tasks, thereby allowing him to finish them faster. Thus, we should also expect that secondary actors can also affect results when the tasks they work on are non-critical, but the primary actor on those tasks also works on critical tasks.

These results suggest that which actor receives the primary task assignment when mutltiple 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.

A more thorough analysis is planned and will analyze the effects of assigning additional people to tasks in ORGAHEAD. The results are shown in Figure 8, but a detailed analysis is just underway. Furthermore, what effect our initial decisions of who the primary actors were had on the results still needs to be determined. By creating the synthetic data set and the experiments just discussed, we have gained a much improved understanding of how secondary actors can affect results and can use this knowledge to drive our analysis of the results that use the real data.

7 Discussion and Conclusion

The two models, SimVision and ORGAHEAD, were tested for equivalence at the organizational level. A generally weaker form of equivalence, "relational equivalence," was used to perform the tests at both levels. Our experiments established a relational equivalence between the models when project duration was used as output in SimVision and accuracy was used as output in ORGAHEAD.

Efforts to align SimVision and ORGAHEAD are not complete. While we did establish relational equivalence, we still need to determine under what conditions this relational equivalence holds. Our approach is to analyze 1) how decisions made during the input of data into the models affect the results, and 2) how and when unique processes and representations in a model affect the results. Thus far, we have gained a much improved understanding of secondary actors in SimVision and how the use of secondary assignments to handle multiple people being assigned to tasks affects results. Other processes will be studied as well.

We have also focused only on a few measurements from the two models. We have collected other measurements for ORGAHEAD that gets at notions of consensus, certainty, time to complete, and ability to complete tasks. It may be the case there are other measurements that can be related between the two models or that the two models can be modified slightly to measure a similar phenomena. In the latter case, the measurement could be designed such that we are able to test for a more powerful equivalence besides relational.

The docking experience has been positive and its continued progress looks promising. The most difficult part has been becoming acquainted with the two models. Until recent, the knowledge I had of how SimVision worked was limited and due only to papers and presentations on the model. Actual use of the model has greatly facilitated my understanding of it. Having good understanding of how both models operate has allowed my analysis to be more directed and removes my dependence on others to run experiments for me in SimVision.

The docking process has been instrumental in our knowledge of how differences in model design can affect the results, even when the models are grounded in the same theories as is the case with the two models studied here. Computational models can be powerful tools for theorizing about organizational dynamics. The confidence people have in them will affect how influential the models are in the field and thus the contributions they can make. Improving understanding among organizational theorists of how the models operate should increase their acceptance as a tool for theorizing. Docking models is a potentially powerful method for gaining a deeper understanding of the models under scrutiny. The results of the docking study can be used to directly increase people's knowledge of the models, but can also help developers of new models by understanding how design choices may affect the results of their models. All in all, we feel the docking process is extremely valuable to both the field of computational organization theory and organization theory as a whole.

Notes

[1] Adaptive conditions showed similar relationships, with the adaptive condition of only permitting changes in connections to have the greatest difference in performance between A06 and A16. The other adaptive conditions follow the same relationship but the difference is not as great.

8 References

Axtel, R., Axelrod R., Epstein, J.M., & Cohen M.D. (1996). Aligning Simulation Models: A Case Study and Results. *Computational and Mathematical Organization Theory*. 1:2, 123-141.

Carley, K.M. (1996). Adaptive organizations: A comparison of strategies for achieving optimal performance. In *Proceedings* of the 1996 International Sysmposium on Command and Control Research and Technology. Monterey, CA.

Carley, K.M. & Svoboda, D.M. (1996). Modeling organizational adaptation as a simulated annealing process. *Sociological Methods and Research*,25(1), 138-168.

Jin, Y. & Levitt, R.E. (1996). The Virtual Design Team: A Computational Model of Project Organizations. *Computational and Mathematical Organization Theory*. 2:3, 171-196.

Kunz, K.C., Levitt, R.E., Jin, Y. (1998). The Virtual Design Team: A Computational Model of Project Organizations. *Communications of the Association for Computing Machinery*. 41(11), 84-92.