

AGENT BASED SIMULATION DESIGN FOR AGGREGATION AND DISAGGREGATION

Tiffany J. Harper
John O. Miller
Raymond R. Hill

J. Robert Wirthlin

Department of Operational Sciences
Air Force Institute of Technology
Wright Patterson AFB, OH 45433, USA

Department of Systems and Engineering Management
Air Force Institute of Technology
Wright Patterson AFB, OH 45433, USA

ABSTRACT

This paper proposes a framework for designing an agent based simulation to allow for easy aggregation and/or disaggregation of agent characteristics, behaviors, and interactions using a supply chain modeling context. Guidelines are provided for designing agent structure to demonstrate scalability in terms of fidelity to fit the needs of the analysis. The design methodology is based on combining hierarchical modeling with data-driven modeling. Related work done in variable-resolution modeling is a generalization for any modeling technique, while our proposed guidelines are specific for development of agent based models.

1 INTRODUCTION

Traditionally, simulation models were used to analyze a specific problem, so model development was rather straight forward and did not require variable levels of detail. With higher complexity systems of today, a single model is often used to analyze a wider spread of problems. Thus, models must have varying levels of fidelity in order to answer the different questions associated with these highly complex systems. Low resolution models are used for initial investigations, comprehension, systems analysis and policy analysis, decision support, adaptability, low cost and rapid analysis, and making use of low-resolution knowledge and data (Davis and Hillestad 1993). High resolution models are used in understanding phenomena, representing knowledge, simulating reality, calibrating or informing lower-resolution models, and making use of high-resolution knowledge and data (Davis and Hillestad 1993).

The concept of using a single model with variable levels of detail, is not new to discrete event simulation (e.g. Davis and Hillestad 1993), but little research exists with a focus on agent based simulation. This paper attempts to layout the process and considerations that go into developing variable fidelity agent-based simulation models. To do this, it is necessary to define terms found in this area of literature. Specifically, we will define and describe the relationship between resolution, scalability, flexibility, and aggregation.

When modeling, resolution can refer to entities, attributes, logical dependency, processes, spatial orientation, or temporal orientation. Table 1 provides military examples of how these six aspects of a model may change with levels of resolution. Granularity, levels of description, and levels of detail are used synonymously for resolution.

Table 1: Aspects of Resolution (Davis and Hillestad 1993)

Aspect of Resolution	Level of Resolution	
	Low	High
Entity	Companies	Battalions
Attribute	Net firepower strength	Number of each weapon system
Logical-dependency	Standard formation	Circumstantial formation
Process	Allocate attrition evenly among battalions on the front line	Compute combat attrition at battalion level based on battle situation
Spatial	Miles	Feet
Temporal	Days	Minutes

Scalability is defined by Rana and Stout (2000) as “the ability of a solution to a problem to work when the size of the problem increases.” Although problem size includes dimensions, such as the data (rules) the agents are operating on (with) and diversity of agents, literature focuses on the number of entities involved.

Flexibility of a simulation refers to it being generic enough to allow for modeling of similar systems by altering the input data used to execute the model (Brown 2010). This concept of using a model for similar systems is referred to as model “re-use” and altering input data for modeling similar systems is known as data-driven modeling.

As defined by the Department of Defense Modeling and Simulation Master Plan (Department of Defense 1995), aggregation is “the ability to group entities while preserving the collective effects of entity behavior and interaction while grouped.” Axtell (1992) defines model aggregation as the decrease in the dimensionality of a simulation model through the fusion of model variables into composite variables. Operators, such as sum, average, minimum, and maximum are the most common form of data and information transformation into an aggregation model (Rodriguez 2008). Aggregation has an inverse relation to resolution. So, as resolution decreases the level of aggregation increases by combining agents and replacing detailed processes with approximations.

Figure 1 depicts the relationship between the various terminology with respect to an aircraft supply chain model. Throughout this paper we will use model resolution when describing levels of model fidelity. To illustrate, an example of a high resolution model with no aggregation is modeling of system performance, while an example of low resolution model with high aggregation is modeling the entire war.

Section 2 provides a literature review. Section 3 presents the standard procedure designing and implementing agent-based modeling and simulation (ABMS). Section 4 proposes guidelines for planning and designing agent structure for handling variable levels of resolution. The proposed guidelines are then demonstrated by a simple example in Section 5, followed by concluding remarks in Section 6.

2 LITERATURE REVIEW

Variable-resolution modeling is defined by Davis and Hillestad (1993) as “building new models or model families so that users can change readily the resolution at which phenomena are treated.” Seamless design refers to designing models such that change in resolution occurs with (a) smooth consistency of representation and (b) consistency of prediction (Davis and Hillestad 1993). In other words, when “zooming” within a model there are no mental disruptions and there is some confidence that the results are consistent. (Davis and Hillestad 1993)

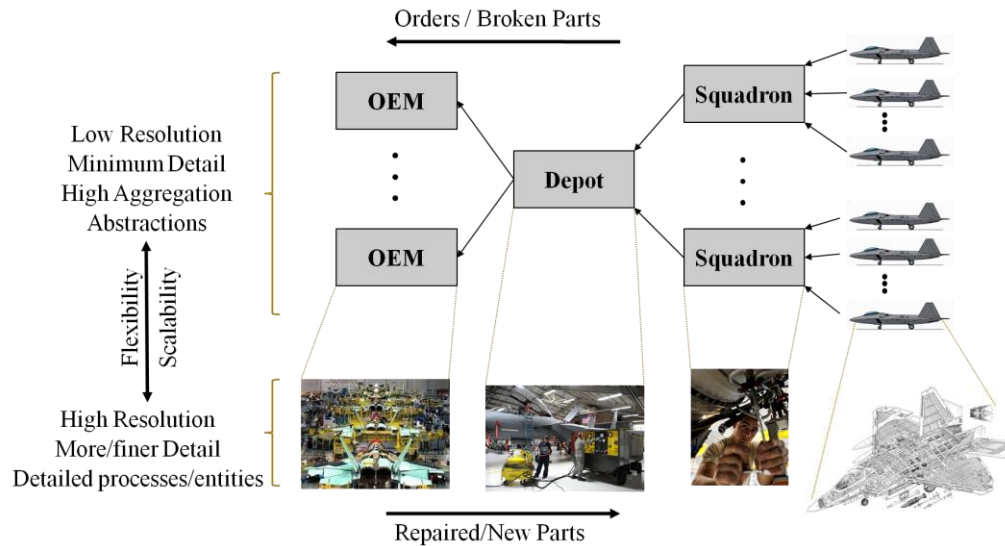


Figure 1: Range of Model Fidelity (Axe 2010; Lockheed Martin 2011; Globalsecurity.org 2011; PACAF 2011; WPAFB 2011)

Three principal approaches can be used to achieve variable resolution modeling, namely, selected viewing, alternative sub models (or model families), and integrated hierarchical variable resolution (IHVR) (Davis and Hillestad 1993). Selected viewing uses the one high resolution model and simply hides logic for low resolution models. The alternative sub models approach consists of different models for levels of resolution and users merely switch to the model corresponding to the desirable level of resolution. IHVR refers to modeling that describes critical processes as being composed hierarchically of subordinate processes and resolution changes by replacing higher-level processes with an approximation, or trivial process, depicted by lookup tables (Davis and Hillestad 1993). The work in variable resolution is focused on discrete-event simulation, but there are smaller studies that focus on resolution complications within ABMS.

To enable scalability and consistency across agents running at different time resolutions, Pawlaszyk and Strassburger (2009) and Chaturvedi et al. (2004) use a gossiping concept. By gossiping among similar time resolution agents and occasionally with differing time resolution agents, the entire system can operate as a whole without running every agent at the smallest time resolution.

Rana and Stout (2000) mitigate coordination issues between diversified agents via an intermediate proxy agent. That is, agents from one group can only communicate to agents of another group through the proxy agent, thus eliminating networking complexity.

The work in this paper is an extension of IHVR to ABMS. Processes are still designed hierarchically of subordinate processes, but the agents are also designed hierarchically. This follows the concept of object-oriented design, where subordinate object classes inherit from higher classes. As with IHVR, lookup tables are used to determine which processes or aggregate processes to use, but with ABMS lookup tables are also used to determine which agents to use and how agents interact.

3 STANDARD ABMS DESIGN METHODOLOGY

As with any simulation study, the first design step is to identify the purpose of the model, the questions the model is intended to answer and the potential users (Macal & North 2005). Then, systematically analyze the system under study, identifying components and component interactions, relevant data sources, and so on (Macal & North 2005). With a basic understanding of the objectives and system under study, the general steps in building an agent-based simulation are depicted by Macal & North (2006) as follows:

1. Agents: Identify the agent types and other objects (classes) along with their attributes
2. Environment: Define the environment the agents will live in and interact with
3. Agent Methods: Specify the methods by which agent attributes are updated in response to either agent-to-agent interactions or agent interactions with the environment
4. Agent Interactions: Add the methods that control which agents interact, when they interact, and how they interact during the simulation
5. Implementation: Implement the agent model in computational software

Normally there is a constant interplay between steps in building an agent-based simulation. Once the development phase is complete the analysis phase is executed, which is typical for general simulation studies. The standard procedure for building and implementing agent-based simulation models is depicted in Figure 2.

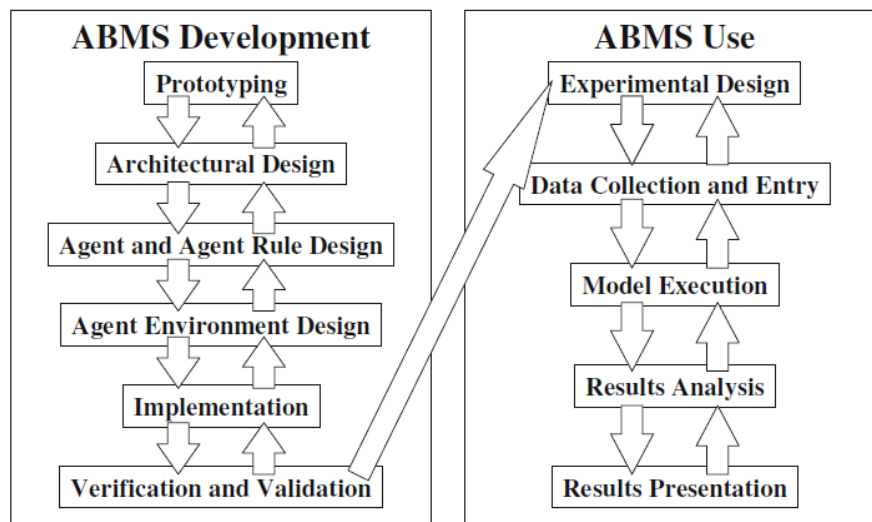


Figure 2: Standard ABMS Procedure (North and Macal 2007)

When ABMS is to be used with different levels of resolution the key steps that are affected are initial planning, agent and agent rule design, data collection and entry, and model execution. While identifying the purpose of the model and the questions the model is intended to answer, there must be some delineation between the different levels or resolution needed for these questions. This is not a trivial process, but can be eased by systematically analyzing the system under study and determining what data is available. As described in Section 4, the way agents are designed will affect the ease of switching levels of resolution. Since multiple levels of resolution have different data requirements, the data collection and entry process is a key step in ABMS for aggregation and disaggregation. More data analysis is necessary to validate the method of data aggregation, so data collection and data analysis will generally take more time than standard ABMS. However, this is balanced by the ability to model and analyze selected parts of the system at a high level of detail or more of the system at an aggregated level. Finally, the model execution process requires some data input changes to change levels of resolution.

4 PLANNING AND DESIGNING AGENTS FOR VARIABLE RESOLUTION

Generic challenges in variable-resolution modeling, as discussed by Davis (1993), include:

- getting the concepts and names straight
- completing sets of variables and functions (i.e. defining the reference model)

- drawing relationships and mappings
- deciding the form of reasonable aggregate equations relative to detailed equations (requires theoretical analysis)
- finding conditions under which aggregation equations might be reasonably valid (requires theoretical analysis)
- expressing aggregate-model parameters in terms of outputs of detailed model (requires theoretical analysis)
- deciding on cases (e.g. scenarios) to be distinguished and how to make calibrations for each case—e.g., how to determine weighting factors over case and time so that calibrations will be appropriate for context of larger applications (requires theoretical analysis)

The two primary issues with changing levels of resolution in an agent-based model are the agents and the processes. At different levels of resolution what agents are active and with what agents do they interact? What processes must be performed on the agents? These are some of the questions that must be asked when planning and designing agents for variable resolution agent-based models.

The basis of the proposed methodology is the combination of hierarchical design with data driven modeling. This method is similar to IHVR by (Davis and Hillestad 1993), but adapted for agent-based modeling. As with IHVR, the proposed methodology utilizes lookup tables and different levels of abstraction for processes, but also for the agents themselves.

4.1 Planning Phase

The planning phase is the most important phase when developing agent-based models with variable resolution. In this phase it is still necessary to identify the purpose of the model, the questions the model is intended to answer and the potential users. However, for variable resolution it is also necessary to delineate between the different levels of resolution needed for the questions to be answered. Specifying the levels of resolution affects what agents are needed and what behaviors and interactions are appropriate. Incorrectly defining the levels of resolution can invalidate and increase difficulty in data collection and building of the agent-based model.

Systematically analyzing the system under study and determining what data is available will aid the process of defining the levels of resolution. Most often availability of data is the key driver in variable resolution modeling. Tools such as process mapping and cause and effect diagrams, along with theory like queuing theory, can also help with determining which details to suppress and which to expand.

Along with planning the agents, agent behavior, interactions and processes simulation input and output must be considered in the initial planning phase. Inputs must be collected to accommodate all levels of resolution and the appropriate aggregation models. A simple method for eliminating the need to change model logic to handle output at different resolutions is to report all outputs. In context to aircraft supply chain, assume aircraft are comprised of engine, landing gear, and body agents. Design the model to collect time to repair data for each agent type (aircraft, engine, landing gear, and body agents). A high resolution model might model failures at the component level (e.g. engine or landing gear) and a low resolution model might aggregate the components into failure of the aircraft. With the high resolution model there will be output data for all agents, while with the low level there will only be output data for aircraft agents. By including all output data you do not have to change the code for levels of resolution.

4.2 Hierarchically Designing Agents

As highlighted by Davis and Hillstad (1993) object-oriented methods can help greatly in developing variable resolution in entities, attributes, and logical-dependency. A key benefit of object-oriented modeling is modularity, which encourages hierarchical representation of objects and attributes (Davis and Hillstad 1993). With object-oriented modeling subclasses inherit attributes (fields) and processes (methods) from higher classes. Many agent-based simulation packages enable hierarchy of objects and processes.

With variable resolution ABMS, different agents, agent behavior, resources, and processes may be necessary. To accommodate this, agents should be defined hierarchically and agent behavior logic should be designed similar to the hierarchical processes depicted in section 6 of (Davis 1993). With hierarchical behavior logic, switches and gates can be employed within the hierarchy to activate the appropriate behavior logic for the corresponding level of resolution. Without designing agents and agent logic hierarchically it would be necessary to manually change large portions of the model to change levels of resolution. For details of how to do hierarchical design with cross-talk between branches and cycling refer to Davis and Huber (1992).

A military example where hierarchy of object-oriented methods would be beneficial is the scenario where a platoon comprised of separate entities encounters an enemy battalion that is modeled as a single entity (Davis and Hillestad 1993). For this scenario the battalion could be disaggregated into separate entities or the platoon's entities could be aggregated into a single entity. With respect to supply chain, an example is modeling depot agents at a low level of resolution and modeling bays, equipment, and personnel agents of a depot at a high level of resolution.

4.3 Designing Agent Interactions

A key problem with variable resolution in ABMS is changing interactions between agents. Hard coding messages between agents could require extensive effort and model changes to switch between various levels of resolution. For a low resolution aircraft supply chain model, broken parts may simply be sent to a squadron for repair whereas the higher resolution model may send broken parts to the flightline or backshops at the squadron. Thus, there is a difference in sending broken parts to the aggregate agent, the squadron, versus sending broken parts to the detailed flightline or backshop. How should agents be designed such that interactions between agents can easily be changed?

Instead of hard coding interactions between agents, lookup tables can be used. With object oriented modeling and systems that do not have individualized interactions the necessary lookup tables are straightforward and changing the tables for different resolutions would not be time consuming. Assume at low resolution all broken parts are sent to squadron agents for repair, but sent to flightlines and backshops at a higher resolution. Then the lookup table for the lower resolution would simply specify to send the message to repair the part to its home squadron, which is an attribute of the agent. Changing to the higher resolution model would only require specifying to send the message to repair the part to its home flightline or backshop. A percentage or condition can also be depicted via table to determine which, flightline or backshop, the message should be sent.

The technique of lookup tables for agent interactions becomes cumbersome when agents of the same type must interact with specific agents of another type. For example, part A can only be repaired at the flightline and part B can only be repaired at the backshop. In this scenario the size of the lookup table would grow rapidly with the number part types and repair locations. With this type of model use of gates and switches in the hard code might be easier. This would enable changing a single variable that links to different switches in the model to accommodate the desired resolution.

If agents and processes are strictly hierarchical, then agent interactions can be inherited from higher classes. That is, if agents in a subclass follow similar processes and interactions as agents in the parent class then the messages can be inherited from the parent class. In the aircraft supply chain example, assume an aircraft gearbox contains a pump, a gear assembly, and a circuit board. If the gearbox is repaired at a home squadron and the pump, gear assembly, and circuit board are also repaired at the home squadron, then a lookup table is not necessary. The message to send the broken part agents to the home squadron can be inherited as a method from the gearbox agent.

4.4 Designing for Aggregate Process Data

Current literature provides numerous aggregation models for processes. Sum, average, minimum, maximum, and mode are some common aggregation models (Rodriguez 2008). Others include regression and

distribution fitting to high resolution model output. These aggregation models can also be used in defining agent behavior.

When aggregating higher resolution processes, theory should first be used to abstract the process. For example, queuing theory. If no theoretical equations are available, then a common aggregation model that could be used is a weighted average of the best, worst, and most likely scenarios. Other common aggregation models, like minimum or maximum, may fit better at this level of aggregation. A final technique of process aggregation is running higher resolution models and fitting regression models and distributions to simulation output. The drawback of this method is a simulation model must already be operational, so model alterations to the existing model might be required to accommodate variable resolution.

Lookup tables are again recommended for implementing the aggregation models in the agent-based model. For a supply chain example, break rates and repair times for individual parts would be specified in the lookup table with higher assemblies having no break rates or repair times. For a lower resolution model the lookup table would have no break rates or repair times for individual parts, but would specify aggregate parameters for the higher assemblies. Lookup tables could be used to specify distributions as well as parameter values. For example, if the process varies over time, then the lookup table would specify what distribution or regression model to be used for the corresponding level of resolution during the specified time period. A similar technique for specifying the distribution or aggregation model is implementation of gates or switches. For different levels of resolution gates/switches can be activated to run the correct aggregation model, that would then utilize the lookup table values.

By planning and designing agents to reference lookup tables, the drawback mentioned previously is eliminated. That is, data for lower resolution models can successively be determined by running the higher resolution models and fitting an aggregate model to the simulation output. Since the agents were designed to reference lookup tables, there is no need to change the existing model.

As with any simulation, the aggregation models and the entire agent-based simulation model must be verified and validated. Standard verification and validation (V&V) methods, such as comparison to historical data and expert assessment, are appropriate at specific levels of resolution in agent-based simulation models. However, variable resolution along with object oriented design introduces complexities and challenges for V&V. For detail on these complexities, challenges and techniques for V&V in the presence of these issues, refer to Balci (1997).

To automate the process of switching between levels of resolution during model execution and analysis phases, lookup tables can be linked to interface controls, such as a slider bar. For example, a slider bar can be coded to specify what agents to implement and change lookup table values according to the specified level of resolution.

A final recommendation for designing agent behavior and process logic is to consider the spatial and temporal orientation. When using decision logic, all time scenarios and spatial orientation must be accommodated. For example, at one level of resolution the model might run in days and all events occur in full days, while a different resolution model might run in hours. If decision logic for the first resolution level uses an equivalence condition alone (e.g. break time = current time, then part breaks), then switching to the resolution with hours will not work correctly because partial days are not considered in the decision logic. To accommodate hours, the logic should implement greater than (less than) along with the equivalence condition (e.g. break time \geq current time, then part breaks). Without the greater than (less than) condition the break event will never trigger. For agents running at different time and spatial orientations refer to Pawlaszczyk and Strassburger (2009) and Chaturvedi et al. (2004).

5 EXAMPLE

To demonstrate the proposed methodology a small theoretical aircraft supply chain model, as depicted in Figure 3, is used for analyzing different repair policies. Assume aircraft landing gear is comprised of two parts, A and B, each with a break rate and repair rate. When a part breaks it is either repaired at the Squadron or sent upstream to the Depot for repair.

Assume the questions of interest are 1) How is aircraft availability affected by increasing the number of parts repaired at the squadron level? and 2) How is aircraft availability affected by increasing the number of landing gear assemblies repaired at the squadron level?

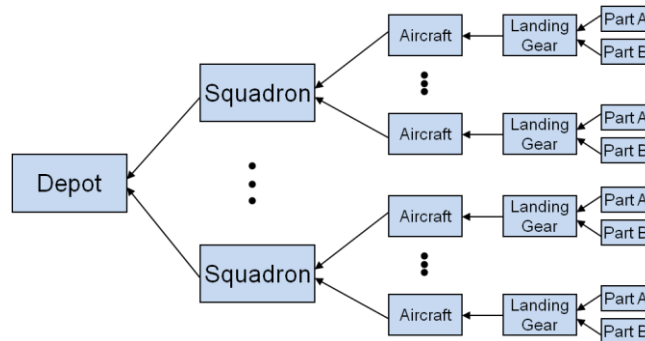


Figure 3: Aircraft Supply Chain Example

The first question requires a high level of detail, where the active agents include Parts, Squadrons, and Depots. Since Landing Gear and Aircraft agents are used simply to track availability output there is no process data for these agents, as depicted in Table 2.

Table 2: Process Parameters for High Resolution Model

	Break Rate	Repair Rate at Squadron	Repair Rate at Depot	Shipment Time to Depot
Part A	15 days	2 days	1 day	2 days
Part B	25 days	5 days	2 days	2 days
Landing Gear	--	--	--	--
Aircraft	--	--	--	--

For the second question the Parts agents are aggregated to become the Landing Gear agents. Thus the lower resolution model for the second question includes Landing Gear agents, Squadrons and Depots. Table 3 shows the process parameters for the active agents for this level of resolution. In this simple example the parameters for the Parts agents were averaged to find the process parameters for the Landing Gear.

Table 3: Process Parameters for Low Resolution Model

	Break Rate	Repair Rate at Squadron	Repair Rate at Depot	Shipment Time To Depot
Part A	--	--	--	--
Part B	--	--	--	--
Landing Gear	20 days	3.5 days	1.5 days	2 days
Aircraft	--	--	--	--

With hierarchical design and object-oriented programming, Aircraft agents form the super class, with successive subclasses Landing Gear agents, then Parts agents. Aircraft agents have six fields, or attributes, that correspond to the data specified in the process parameter tables. Figure 4 provides pseudo code for defining these agents hierarchically with object-oriented programming.


```
Class Aircraft {
  //Aircraft agents have 6 fields
  Home Squadron //Squadron to be repaired at
  Home Depot //Depot to be repaired at
  Break Rate = Aircraft Break Rate
  Squadron Repair Rate = Aircraft Repair Rate at Squadron
  Depot Repair Rate = Aircraft Repair Rate at Depot
  Shipment Time to Depot = Aircraft Shipment Time to Depot

  //Aircraft agents have 2 Methods
  Break {
    Time of Break = Current Time
    Send message to Home Squadron for Repair
  }
  Repaired {
    Time to Repair = Current Time – Time of Break //tracks the time to repair the agent
    Time Operational = Total Simulation Time – Time to Repair each break
    Availability = Total Simulation time / Time operational
  }
}
Class Landing Gear extends Aircraft {
  //Landing Gear is a subclass of Aircraft so it inherits the 6 fields and 2 methods from Aircraft
  //The process data is overridden for Landing Gear agents
  Break Rate = Landing Gear Break Rate
  Squadron Repair Rate = Landing Gear Repair Rate at Squadron
  Depot Repair Rate = Landing Gear Repair Rate at Depot
  Shipment Time to Depot = Landing Gear Shipment Time to Depot
}
Class Parts extends Landing Gear {
  //Part is a subclass of Landing Gear so it inherits the 6 fields and 2 methods from Landing Gear
  //The process data is overridden for Part agents
  Break Rate = Part Break Rate from
  Squadron Repair Rate = Part Repair Rate at Squadron
  Depot Repair Rate = Part Repair Rate at Depot from
  Shipment Time to Depot = Part Shipment Time to Depot
}
```

Figure 4: Aircraft Supply Chain Example

To demonstrate the use of lookup tables for agent interactions, the same aircraft supply chain example is used to answer questions regarding repair processes at the Squadron level. Assume the questions of interest are now 1) How is aircraft availability affected by repairing more parts on the flightline versus repairing parts in the backshops? and 2) How is aircraft availability affected by increasing the number of parts repaired at the Squadron?

Table 4 shows the agent interactions for repairing the Part agents at the Flightline and Backshop level, which is the higher resolution model. For the lower resolution model in the second question, Part agents interact with the Squadron agents, as depicted in the Table 5.

Table 4: Agent Interactions for High Resolution Model

	Repair Message Sent To	Repaired Message Sent To
Parts (A, B)	Flightline / Backshop	--
Landing Gear	--	--
Aircraft	--	--
Squadron	--	--
Flightline	Depot	Parts (A, B)
Backshop	Depot	Parts (A, B)
Depot	EOM	Flightline / Backshop

Table 5: Agent Interactions for Low Resolution Model

	Repair Message Sent To	Repaired Message Sent To
Parts (A, B)	Squadron	--
Landing Gear	--	--
Aircraft	--	--
Squadron	Depot	Parts (A, B)
Flightline	--	--
Backshop	--	--
Depot	EOM	Squadron

As mentioned previously, in the simple case where all parts have the same logic gates and switches can be used instead of lookup tables.

6 CONCLUSION

By combining hierarchical modeling with data-driven modeling the proposed methodology has extended the variable resolution modeling work to agent-based modeling and simulation (ABMS). This work ties together a general framework for using ABMS for supply chain risk management, which includes the use of software agents, for data mining, integrated with agent-based simulation platforms. This framework enables rapid data collection for simulation input, while also providing an intuitive simulation platform.

REFERENCES

- Axe, D. 2010. "Lockheed Cross-Breeding Raptors, Joint Strike Fighters." *Wired*. Accessed May 10, 2011. <http://www.wired.com/dangerroom/2010/12/lockheed-cross-breeding-raptors-joint-strike-fighters/>.
- Axtell, R. L. 1992. "Theory of Model Aggregation for Dynamical Systems with Application to Problems of Global Change." Ph.D. Dissertation. Carnegie-Mellon University.
- Balci, O. 1997. "Verification, Validation and Accreditation of Simulation Models." In *Proceedings of the 1997 Winter Simulation Conference*, edited by S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson, 135-141. Piscataway, New Jersey: Institute of Electronics Engineers, Inc.
- Brown, N. 2010. "Model Flexibility: Development of a Generic Data-Driven Simulation." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hukan, and E. Yucesan, 1366-1375. Piscataway, New Jersey: Institute of Electronics Engineers, Inc.
- Chaturvedi, A., J. Chi, S. Mehta, and D. Dolk. 2004. "SAMAS: Scalable Architecture for Multi-resolution Agent-Based Simulation." *Lecture Notes in Computer Science* 3038: 779-788.

- Davis, P. 1993. "An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection." RAND R-4252-DARPA, Santa Monica, CA.
- Davis, P. and R. Hillestad. 1993. "Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management." In *Proceedings of the 1993 Winter Simulation Conference*, edited by G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 1003-1012. Piscataway, New Jersey: Institute of Electronics Engineers, Inc.
- Davis, P. and R. Huber. 1992. "Variable-Resolution Combat Modeling: Motivations, Issues, and Principles." RAND N-3400-DARPA, Santa Monica, CA.
- Department of Defense. 1995. "Department of Defense Modeling and Simulation Master Plan." DoD Directive 5000.59-P. Washington: GPO, 1995.
- Globalsecurity.org. 2011. F-22 Raptor Pictures. Accessed April 14. <http://www.globalsecurity.org/military/systems/aircraft/f-22-pics.htm>.
- Lockheed Martin. 2011. Lockheed Martin F-22 Raptor diagram. Accessed June 1. <http://www.aerospaceweb.org/aircraft/fighter/f22/pics06.shtml>.
- Macal, C. M. and M. J. North. 2005. "Tutorial on Agent-Based Modeling and Simulation." In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2-15. Piscataway, New Jersey: Institute of Electronics Engineers, Inc.
- Macal, C. M. and M. J. North. 2006. "Tutorial on Agent-Based Modeling and Simulation Part 2: How To Model With Agents." In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 73-83. Piscataway, New Jersey: Institute of Electronics Engineers, Inc.
- North, M. J. and C. M. Macal. 2007. "Tutorial on Agent-based Modeling and Simulation." *75th Military Operations Research Society Symposium (MORSS)*. Annapolis, MD.
- PACAF. 2011. Pacific Air Forces. Accessed June 5. http://www.pacaf.af.mil/photos/media_search.asp?q=f-22%20f-22s%20f-22a&page=6.
- Pawlaszczyk, D. and S. Strassburger. 2009. "Scalability in Distributed Simulations of Agent-Based Models." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 1189-1200. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Rana, O. F. and K. Stout. 2000. "What is Scalability in Multi-Agent Systems." In *Proceedings of the fourth International Conference on Autonomous Agents*. 56-63.
- Rodriguez, J. F. D. 2008. *Metamodeling Techniques to Aid in the Aggregation Process of Large Hierarchical Simulation Models*. Dissertation: Air Force Institute of Technology.
- WPAFB. 2011. Wright Patterson Air Force Base. Accessed May 30. <http://www.wpafb.af.mil/photos/mediagallery.asp?galleryID=2600&?id=-1&page=1&count=48>.

AUTHOR BIOGRAPHIES

TIFFANY J. HARPER holds a B.S. and M.S. in Industrial and Systems Engineering at Wright State University and is currently a Ph.D. student in the Department of Operational Sciences at the Air Force Institute of Technology. Her interests include applications of modeling and simulation. She is also presently working with Riverside Research. Her email address is tharper@afit.edu.

JOHN O. MILLER is a 1980 graduate of the U.S. Air Force Academy (USAFA) and retired from the Air Force as a Lt. Colonel in January 2003. In addition to his undergraduate degree from USAFA, he received an MBA from the University of Missouri at Columbia in 1983, his M.S. in Operations Research from the Air Force Institute of Technology (AFIT) in 1987, and his Ph.D. in Industrial Engineering from The Ohio State University in 1997. Dr. Miller is the Deputy Department Head and an associate professor

in the Department of Operational Sciences at AFIT. His research interests include combat modeling, computer simulation, and ranking and selection. His email address is jmiller@afit.edu.

J. ROBERT WIRTHLIN is a 1994 graduate of the U. S. Air Force Academy (USAFA) and is currently a Lt. Colonel in the United States Air Force. He received his Master's degree from MIT in Systems Design and Management in 2000 and his PhD from MIT in Engineering Systems in 2009. He is an Assistant Professor of Engineering Systems at the Air Force Institute of Technology and the Program Director of the Research and Development Graduate Degree Program. He is also a Visiting Scholar with the USAF Center for Systems Engineering. His research interests include new product development of complex systems, risk, and application of lean to product development. His email address is jwirthli@afit.edu.

RAYMOND R. HILL is a Professor of Operations Research in the Department of Operational Sciences at the Air Force Institute of Technology. He is a retired Air Force Lieutenant Colonel and holds a PhD in Industrial and Systems Engineering from The Ohio State University. His research interests are in the areas of applied statistics and experimental design, modeling and simulation, heuristic optimization and military aspects of modeling and analysis. He is the General Chair for the 2013 Winter Simulation Conferences. His email address is rayrhill@gmail.edu.