

Athena User's Guide

Athena S&RO Simulation, V3

February, 2012

William H. Duquette
Jet Propulsion Laboratory

Copyright 2008-2012, by the California Institute of Technology. ALL RIGHTS RESERVED. United States Government Sponsorship acknowledged. Any commercial use must be negotiated with the Office of Technology Transfer at the California Institute of Technology.

This software is subject to U.S. export control laws and regulations and has been classified as EAR99. By accepting this software, the user agrees to comply with all applicable U.S. export laws and regulations. User has the responsibility to obtain export licenses, or other export authority as may be required before exporting such information to foreign countries or providing access to foreign persons.

Table of Contents

| | |
|---|----------|
| PART I: OVERVIEW..... | 7 |
| 1. Introduction..... | 8 |
| 1.1 Overview of This Document..... | 8 |
| 1.2 Other Documents..... | 8 |
| 1.3 Changes for Athena 3 | 9 |
| 2. Athena Model Overview | 11 |
| 2.1 The Simulation and its Objects..... | 11 |
| 2.2 The Six Modeling Areas..... | 12 |
| 3. Ground..... | 14 |
| 3.1 Simulated Time..... | 14 |
| 3.2 The Playbox..... | 14 |
| 3.2.1 Neighborhoods..... | 14 |
| 3.2.2 Neighborhood Proximity..... | 15 |
| 3.2.3 Neighborhood Effects Delay | 15 |
| 3.2.4 Local vs. Non-Local Neighborhoods | 15 |
| 3.2.5 Production Capacity | 16 |
| 3.3 Actors..... | 16 |
| 3.4 Groups | 17 |
| 3.4.1 Civilian Groups..... | 17 |
| 3.4.2 Force Groups..... | 17 |
| 3.4.3 Organization Groups | 18 |
| 3.5 Deployment..... | 18 |
| 3.6 Activity Assignment..... | 19 |
| 3.7 Units..... | 19 |
| 3.8 Volatility and Security | 19 |
| 3.9 Coverage | 21 |
| 3.10 Activity Situations | 21 |
| 3.11 Environmental Situations | 22 |
| 3.12 Athena Attrition Model | 22 |
| 3.12.1 Rules of Engagement..... | 23 |
| 3.12.2 Presence and Intelligence..... | 23 |
| 3.12.3 Attrition Assessment..... | 23 |
| 3.12.4 Magic Attrition | 24 |
| 3.13 Essential Non-Infrastructure (ENI) Services..... | 24 |
| 3.13.1 The Notion of a Service | 24 |
| 3.13.2 Services vs. Environmental Situations | 26 |
| 3.13.3 Measurement of ENI Services..... | 25 |
| 3.13.4 Required Level of ENI Services..... | 26 |
| 3.13.5 Status Quo ENI Funding..... | 26 |
| 3.13.6 Effects of ENI Services | 26 |
| 4. Demographics | 28 |
| 4.1 Base Population..... | 28 |
| 4.2 Current Population | 28 |
| 4.3 Subsistence Agriculture | 28 |
| 4.4 Consumers and Workers..... | 29 |
| 4.5 Demographic Situations | 29 |
| 5. Attitudes..... | 30 |
| 5.1 Belief Systems and Affinities | 30 |
| 5.1.1 Beliefs and Topics..... | 30 |
| 5.1.2 Affinity..... | 32 |
| 5.1.3 Playbox Commonality..... | 32 |
| 5.1.4 Entity Commonality | 32 |

| | | |
|-----------------------------------|---|-----------|
| 5.2 | Horizontal Relationships..... | 33 |
| 5.2.1 | Relationship Overrides | 33 |
| 5.3 | Vertical Relationships..... | 33 |
| 5.3.1 | Force and Organization Groups | 34 |
| 5.3.2 | Civilian Groups..... | 34 |
| 5.4 | Satisfaction Levels..... | 34 |
| 5.4.1 | The Four Concerns | 35 |
| 5.4.2 | Saliencies | 35 |
| 5.4.3 | Group Mood..... | 35 |
| 5.5 | Cooperation Levels | 36 |
| 5.6 | Trends and Thresholds | 36 |
| 5.7 | The Driver Assessment Model (DAM)..... | 36 |
| 5.7.1 | Level and Slope Inputs | 37 |
| 5.7.2 | Direct and Indirect Effects..... | 37 |
| 5.8 | Magic Attitude Drivers | 37 |
| 6. | Politics..... | 38 |
| 6.1 | Strategies: Goals, Tactics, and Conditions..... | 38 |
| 6.1.1 | Assets..... | 38 |
| 6.1.2 | Conditions | 39 |
| 6.1.3 | Goals..... | 39 |
| 6.1.4 | Tactics | 39 |
| 6.1.5 | Strategy Execution | 40 |
| 6.1.6 | What the Actor Knows, and When He Knows It..... | 40 |
| 6.2 | Support, Influence, and Control | 41 |
| 6.2.1 | Support | 41 |
| 6.2.2 | Influence..... | 42 |
| 6.2.3 | Control..... | 42 |
| 6.2.4 | When Control Shifts..... | 43 |
| 7. | Economics..... | 44 |
| 7.1 | Dollars..... | 44 |
| 7.2 | Sectors | 44 |
| 7.3 | Shape vs. Size..... | 45 |
| 7.4 | Economic Outputs | 45 |
| 7.5 | Neighborhood Aggregation/Disaggregation | 46 |
| 7.5.1 | Neighborhood Aggregation..... | 46 |
| 7.5.2 | Neighborhood Disaggregation | 46 |
| 7.6 | Ways to Affect the Economy | 47 |
| 7.7 | Effects of the Economy | 47 |
| 8. | Information..... | 48 |
| PART II: USING ATHENA..... | | 49 |
| 9. | Installation..... | 50 |
| 9.1 | Starting Athena | 50 |
| 9.2 | Athena Documentation | 50 |
| 9.3 | Multiple Versions of Athena..... | 50 |
| 10. | Using the Athena Application..... | 51 |
| 10.1 | Athena Scenario Files..... | 51 |
| 10.2 | Athena Workflow..... | 51 |
| 10.3 | Scenario Mode vs. Simulation Mode..... | 52 |
| 10.4 | Viewing Athena Results | 52 |
| 10.5 | The Significant Events Log | 53 |
| 10.6 | The Debugging Log | 53 |
| 10.7 | Background Errors..... | 53 |
| 10.8 | Athena Scripting..... | 54 |
| 10.8.1 | Executive Commands..... | 54 |
| 10.8.2 | The Athena Command Line..... | 54 |

| | | |
|---------|--|----|
| 10.8.3 | Executive Command Scripts | 54 |
| 10.8.4 | Scenario Scripts | 55 |
| 10.9 | Batch Mode | 55 |
| 10.9.1 | Invoking Athena in Batch Mode | 55 |
| 10.9.2 | Simulation Control | 56 |
| 10.9.3 | Simulation Results | 56 |
| 11. | The Athena User Interface | 58 |
| 11.1 | The Main Window | 59 |
| 11.2 | The Menu System | 60 |
| 11.2.1 | The File Menu | 60 |
| 11.2.2 | The Edit Menu | 60 |
| 11.2.3 | The View Menu | 60 |
| 11.2.4 | The Orders Menu | 61 |
| 11.2.5 | The Reports Menu | 61 |
| 11.2.6 | The Help Menu | 61 |
| 11.3 | The Main Toolbar | 62 |
| 11.3.1 | The Scenario Mode Toolbar | 62 |
| 11.3.2 | The Simulation Mode Toolbar | 62 |
| 11.4 | The Command Line | 63 |
| 11.5 | The Detail Browser | 64 |
| 11.5.1 | Detail Browser Sidebar | 64 |
| 11.5.2 | Detail Browser Search Box | 65 |
| 11.5.3 | Context Menu | 65 |
| 11.5.4 | Useful URLs | 65 |
| 11.6 | The Map Browser | 67 |
| 11.6.1 | The Map Proper | 67 |
| 11.6.2 | Map Reference Strings | 67 |
| 11.6.3 | Neighborhood Polygons | 68 |
| 11.6.4 | Unit and Environmental Situation Icons | 68 |
| 11.6.5 | Map Browser: Main Toolbar | 69 |
| 11.6.6 | The Secondary Toolbar | 69 |
| 11.6.7 | Context Menus | 70 |
| 11.7 | The Strategy Browser | 71 |
| 11.7.1 | The Agent List | 71 |
| 11.7.2 | The Goal/Condition Pane | 71 |
| 11.7.3 | The Tactic/Condition Pane | 72 |
| 11.8 | The Belief System Browser | 74 |
| 11.8.1 | The Topics Pane | 74 |
| 11.8.2 | The Entities Tree | 75 |
| 11.8.3 | The Beliefs Pane | 76 |
| 11.8.4 | The Affinities Pane | 76 |
| 11.9 | The Neighborhoods Tab | 77 |
| 11.10 | The Groups Tab | 79 |
| 11.11 | The Attitudes Tab | 81 |
| 11.12 | The Demog Tab | 83 |
| 11.13 | The Econ Tab | 84 |
| 11.14 | The Plots Tab | 85 |
| 11.14.1 | Plot Browser Toolbar | 85 |
| 11.14.2 | Plot Toolbar | 86 |
| 11.14.3 | Plot Context Menu | 86 |
| 11.15 | The Firings Tab | 87 |
| 11.16 | The Orders Tab | 89 |
| 11.17 | The Log Tab | 90 |
| 12. | Creating an Athena Scenario | 91 |
| 12.1 | The Actors | 91 |
| 12.2 | The Map | 91 |
| 12.3 | The Neighborhoods | 92 |

| | | |
|---------------------------------------|---|------------|
| 12.4 | Neighborhood Proximities | 92 |
| 12.5 | Civilian Groups..... | 94 |
| 12.6 | Force Groups..... | 94 |
| 12.7 | Organization Groups | 94 |
| 12.8 | Belief Systems | 95 |
| 12.8.1 | Define the Topics | 95 |
| 12.8.2 | Define the Beliefs | 95 |
| 12.8.3 | Compute the Affinities..... | 96 |
| 12.8.4 | Adjust the Affinities | 96 |
| 12.9 | Horizontal Relationships..... | 96 |
| 12.10 | Group Satisfaction | 97 |
| 12.11 | Group Cooperation..... | 97 |
| 12.12 | Status Quo Deployments | 98 |
| 12.13 | Status Quo ENI Funding..... | 98 |
| 12.14 | Environmental Situations | 98 |
| 12.15 | Strategies..... | 99 |
| 12.15.1 | The Role of an Actor's Strategy | 99 |
| 12.15.2 | Order Matters..... | 99 |
| 12.15.3 | Use of Goals | 99 |
| 12.16 | The Model Parameter Database..... | 100 |
| 12.16.1 | Model Parameters..... | 100 |
| 12.16.2 | Browsing the Model Parameter Database..... | 100 |
| 12.16.3 | Parameter Help | 101 |
| 12.16.4 | Editing Model Parameters | 101 |
| 12.16.5 | Model Parameter Files..... | 101 |
| PART III: ATHENA COOKBOOK..... | | 103 |
| 13. | Recipes | 104 |
| 13.1 | Disabling the Economic Model | 104 |
| 13.2 | Calibrating the Economic Model..... | 104 |
| 13.3 | Regression of an Attitude to a Mean | 104 |
| PART IV: REFERENCE | | 106 |
| 14. | Athena Objects..... | 107 |
| 14.1 | Neighborhoods | 108 |
| 14.2 | Actors..... | 109 |
| 14.3 | Civilian Groups..... | 110 |
| 14.4 | Force Groups..... | 111 |
| 14.5 | Organization Groups | 112 |
| 14.6 | Satisfaction Levels..... | 113 |
| 14.7 | Cooperation Levels | 114 |
| 14.8 | Environmental Situations | 115 |
| 14.8.1 | Environmental Situation Types | 115 |
| 14.8.2 | Operations on Environmental Situations..... | 116 |
| 14.8.3 | Attributes of Environmental Situations..... | 117 |
| 15. | Tactic Types | 118 |
| 15.1 | ASSIGN..... | 119 |
| 15.1.1 | Activities..... | 119 |
| 15.2 | ATTROE..... | 121 |
| 15.3 | DEFROE..... | 122 |
| 15.4 | DEMOB | 123 |
| 15.5 | DEPLOY | 124 |
| 15.6 | DISPLACE | 125 |
| 15.7 | EXECUTIVE | 126 |
| 15.8 | FUND | 127 |

| | | |
|-------|--------------------------|-----|
| 15.9 | FUNDENI..... | 128 |
| 15.10 | MOBILIZE | 129 |
| 15.11 | SAVE | 130 |
| 15.12 | SPEND | 131 |
| 15.13 | SUPPORT | 132 |
| 16. | Condition Types | 133 |
| 16.1 | AFTER | 134 |
| 16.2 | AT | 135 |
| 16.3 | BEFORE | 136 |
| 16.4 | CASH | 137 |
| 16.5 | CONTROL..... | 138 |
| 16.6 | DURING | 139 |
| 16.7 | EXPR | 140 |
| 16.8 | INFLUENCE..... | 141 |
| 16.9 | MET | 142 |
| 16.10 | MOOD | 143 |
| 16.11 | NBCOOP | 144 |
| 16.12 | NBMOOD..... | 145 |
| 16.13 | TROOPS | 146 |
| 16.14 | UNMET | 147 |
| 17. | Simulation Orders | 148 |
| 18. | Glossary | 149 |
| 19. | Acronyms | 166 |
| 20. | Scenario Checklist | 168 |

Part I: Overview

1. INTRODUCTION

This document presents the models and software of the Athena 3.1 Stability & Recovery Operations (S&RO) Simulation from the user's point of view. Users are advised to read this document before moving on to the other Athena documents.

The Athena simulation is a decision support tool designed to allow a skilled analyst to consider the intended and unintended consequences of various courses of action that might be taken during Stability & Recovery Operations. Athena contains models descended from the Joint Non-kinetic Effects Model (JNEM), and includes many new models and other changes. In addition, where JNEM is a federated simulation, Athena is a stand-alone single-user application.

The intent of Athena's models is first to capture and make explicit a wide variety of first order causal links, each of which makes sense on the face of it, and secondly to present the second and third order consequences of events while preserving the causal chain.

Everyone is familiar with the story of the six blind men and the elephant. The goal of Athena is to model each of the elephant's parts, and to link them together so that the man who has the elephant by the tail is sure to get thwacked by the elephant's trunk (not to mention everything in between) and so must pay attention to the entire elephant.

1.1 Overview of This Document

This document covers four major topics.

- Part I (Sections 1 through 8) describes the Athena models and philosophy at a conceptual level; those interested in more detail can see the low-level model descriptions in the *Athena Analyst's Guide* and other documents.
- Part II (Sections 9 through 12) describes the Athena application itself: the parts of the application, how to enter scenario data, how to run the simulation, and how to find the results.
- Part III (Section 13) contains a cookbook of how to make use of Athena's models and inputs for particular problems.
- Part IV (Sections 14 through 20) contains reference information, including a complete glossary of terms. Detailed reference information can be found in the Athena application's on-line help.

1.2 Other Documents

In addition to this user's guide, Athena is delivered with the following documents:

Athena Analyst's Guide

This document contains a detailed, low-level description of the models used in Athena; it serves as the specification document for the implementation of the models in the Athena code. Consult it when you need to know more about the models than is contained in this user's guide.

Mars Analyst's Guide

Mars is an infrastructure layer that is shared with the Joint Non-kinetic Effects Model (JNEM). The *Mars Analyst's Guide* can be thought of as an appendix to the *Athena Analyst's Guide* that describes the models that happen to be implemented in the Mars code base, including the Generalized Regional Attitude Model (GRAM) and the Mars Affinity Model (MAM). The former tracks direct and indirect effects on civilian attitudes; the latter models belief systems and the resulting affinities between actors and groups.

Athena Rules Document

This document describes the events and situations (drivers) that affect civilian attitudes in more detail than does this user's guide, and also details each of the Driver Assessment Model (DAM) rule sets that assess the attitude change caused by the drivers.

Athena On-line Help

The Athena application includes extensive on-line help; see the **Help** menu in the application's main menu bar. The most detailed reference information for the Athena software is found in the on-line help.

When Athena is installed on Microsoft Windows, these documents are available from the Athena folder on the Start Menu. Alternatively, go to the Athena application directory and open "docs\index.html" in a web browser. Documentation can also be obtained directly from the Athena Project; contact William.H.Duquette@jpl.nasa.gov.

1.3 Changes for Athena 3

In its conception, Athena was intended to be a single-user version of JNEM customized and extended to be a decision support tool for courses of action in the S&RO environment. JNEM depended on an external federation of simulations for much of its simulation input, e.g., combat, civilian casualties, presence and location of military forces, and so forth. Athena versions 1 and 2 relied on the analyst for much of this input as the simulation ran. In particular, the analyst was expected to run Athena forward in short one month to three month time steps; at each pause, the analyst was to role play each of the relevant actors in the region, make appropriate inputs and adjustments, and then advance time again. This placed a great burden on the analyst.

The big change in Athena 3 is the addition of actors and their strategies. By defining the relevant actors and their strategies, the analyst can set up a complete scenario, and then let the modeled actors respond to the changing conditions. It is still possible for the analyst to pause frequently and make course-corrections if desired, but this is no longer an essential feature.

More specifically, in Athena 3 we have:

- Added actors and their strategies (goals, tactics, and attached conditions)
- Added the notion of *belief systems*; comparison of belief systems is the basis for the model of inter-group and group/actor relationships.
- Added a model of actor support, influence, and control of neighborhoods.
- Added a model of Essential Non-Infrastructure Services, which uses a new paradigm for driving attitude change.
- Revised the user interface:
 - Separated Scenario Mode from Simulation Mode. The data presented is now appropriate to the task being done. In particular, data populated only during simulation is now longer visible during scenario preparation.
 - Added the Detail Browser, a web-browser-like window for browsing the scenario and the simulation results.

2. ATHENA MODEL OVERVIEW

Athena is a collection of many models that involve the relations and interactions between a several kinds of simulation object. This section gives a brief overview of the most important kinds of simulation object, and of the six major modeling areas. The various kinds of simulation object are documented in detail in Section 14; the modeling areas and the models they contain are described more fully in Sections 3 through 8, and are documented in detail in the *Athena Analyst's Guide*, *Mars Analyst's Guide*, and *Athena Rules* documents.

2.1 The Simulation and its Objects

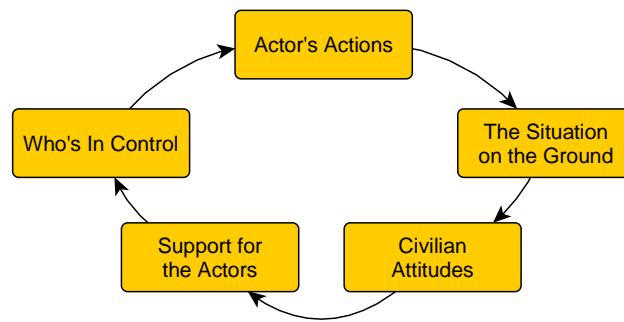
At its highest level, Athena models the actions taken by significant decision makers, called *actors*, within a region of interest, called the *playbox*. The playbox is divided into sub-regions, called *neighborhoods*. Each neighborhood is inhabited by some number of civilians, who are divided into *civilian groups*, and who form both the consumers and the labor force in the *local economy*.

The actors attempt to control, aid, or otherwise influence the civilians in the neighborhoods by means of their actions, which are called *tactics*. Executing tactics requires *assets*, of which there are two kinds: money and personnel. Each actor has an income, and may also own bodies of personnel, e.g., army troops, police forces, humanitarian relief organizations, and so forth. These are called *force groups* or *organization groups*, depending on the nature of the group. By attaching *conditions* to tactics, an actor can determine when and under what circumstances tactics are used. The collection of an actor's tactics and conditions is called the actor's *strategy*.

The civilian groups have *attitudes*: satisfaction or dissatisfaction with respect to particular, and a willingness or unwillingness to cooperate (i.e., share information with) members of force groups. These attitudes vary over time in response to the events and situations that occur in the simulation, including those triggered by the actors' actions.

In addition, civilian groups can *support* actors to a greater or lesser degree. Support is based upon shared or compatible beliefs, but is also affected by actors' actions and by conditions in the civilians' neighborhoods. Members of force and organization groups also support the actor to whom the group belongs. Actors may use the support they receive, or lend it to other actors. An actor with sufficient support in a neighborhood is said to have *influence* in the neighborhood; and an actor with sufficient influence may *control* the neighborhood.

Thus, we have the following feedback loop:



The actors' actions determine the situation on the ground, which in turn affects civilian attitudes. This in turn affects civilian support for the actors, which can cause a change in which actor controls each neighborhood. That political situation then drives the actors' actions. There are, of course, smaller feedback loops within this one; for example, an actor can increase his support in a neighborhood by moving a significant military force into that neighborhood. The force supports him, possibly enough so that he gains control, bypassing (at least temporarily) civilian attitudes altogether. Similarly, there is a feedback loop from the situation on the ground back to the actors' actions—even politicians look at more than just the latest polls. But at a high level, this is the dynamic that drives Athena execution.

The primary outputs at the end of an Athena run are these:

- The actor in control of each neighborhood
- The stability and security of each neighborhood
- The civilian groups' attitudes about the state of the playbox
- The resources decisions required by each actor to bring about this end state.

Thus, Athena can be used both to analyze an existing political situation and to assess the results of various courses of action designed to change it, from the point of view of any or all of the actors involved.

2.2 The Six Modeling Areas

Athena's models fall loosely into six broad areas; the borders between these areas are often fuzzy, and some of the Athena models straddle them. The six areas are as follows:

The *Ground* Area

Literally, where things are on the ground, the things they are doing, and what is happening to them as a result. (Section 3)

The *Demographics* Area

This area concerns where civilians live and how many of them there are, along with the computation of such population statistics as the number of consumers in the local economy. (Section 4)

The *Attitudes* Area

This area deals with the attitudes of the people in the playbox, especially their satisfaction and cooperation levels, but also their belief systems and the relationships between the various groups in the playbox. (Section 5)

The *Politics* Area

This area deals with actors and their strategies (goals, tactics, and attached conditions) along with the determination of support, influence, and neighborhood control. (Section 6)

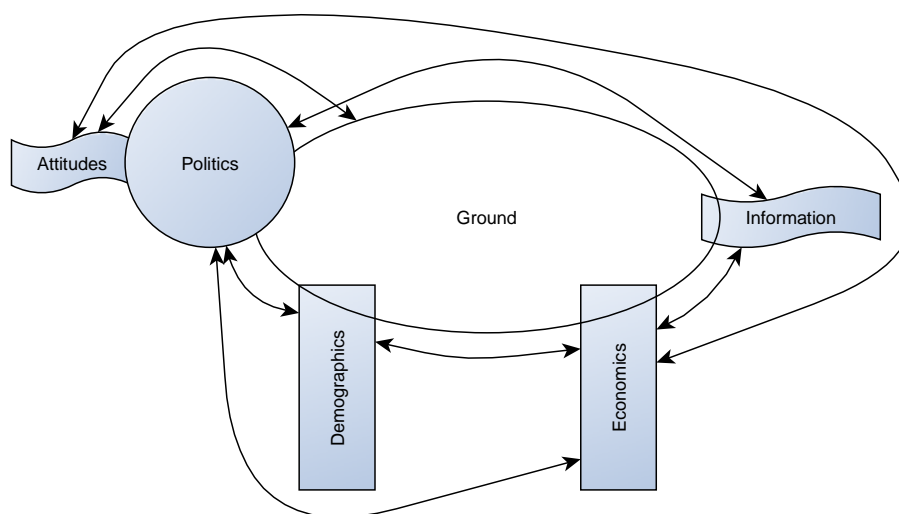
The *Economics* Area

This area deals with the local economy, relating changes in population and production to the consumer price index (CPI) and the unemployment rate. (Section 7)

The *Information* Area

This area deals with information flow in the playbox, and especially the effect of information flow and information campaigns on attitudes and politics. (Section 8)

The models in each of these areas are inter-related: every area depends on inputs and outputs from the other areas, as shown in the following figure. It is necessary to track all of them to get a complete view of the simulation.



Each of these areas and the models contained within it are described in the referenced sections; for full details on the models, see the *Athena Analyst's Guide*, *Mars Analyst's Guide*, and *Athena Rules* documents.

3. GROUND

The Ground Area is quite literally concerned with what is happening on the ground: where people are, and what they are doing, and the results of their actions. It includes:

- The breakdown of the playbox into neighborhoods and the relationships among the neighborhoods.
- Where civilian, force, and organization personnel are located
- The activities that they are performing, including combat activities
- Neighborhood security levels
- Activity coverage
- Environmental situations
- Essential Non-Infrastructure (ENI) services
- The Athena Attrition Model (AAM)
- Where production capacity is located.

Events and situations occurring in the Ground Area affect civilian attitudes, and the demographics and production capacity of the playbox and hence the economy as well. Actors base their decisions on the state of affairs in the playbox.

3.1 Simulated Time

Athena measures the passage of time in integer days. It is a time-step simulation, with a step-size of one day; however, most simulated happenings take place week-by-week, e.g., combat attrition is assessed once each week.

3.2 The Playbox

The playbox is the geographic area in which the simulation takes place. It is modeled as a collection of polygonal regions called *neighborhoods*, which are laid out on a map. In Athena 3, the neighborhoods are simply convenient bins for collecting simulation objects that are near each other; the layout of neighborhood polygons on a map is simply an aid to visualization.

3.2.1 Neighborhoods

Almost everything that happens in Athena takes place in the context of a neighborhood. Civilian groups reside in neighborhoods; force and organization group personnel are deployed to neighborhoods; personnel and actors act in neighborhoods; attitudes and output statistics are measured in neighborhoods.

Neighborhoods can be of any size, from portions of a city (neighborhoods in the proper sense of the word) to entire cities, counties, districts, provinces, countries, or groups of countries.

Neighborhoods can nest, i.e., a neighborhood representing a city can be placed on top of a neighborhood representing a province.

Each neighborhood can be more or less urbanized, and can contain more or less of the playbox's economic production capacity.

3.2.2 Neighborhood Proximity

Neighborhoods are simply bins; the geographic layout of the neighborhoods on the map has no effect, in and of itself, on the simulation results. It is clear, however, that some neighborhoods are more closely related than others. Because this is a social distance, not a physical distance, we model it directly rather than deriving it from the geography.

We call this social distance *neighborhood proximity*; it is defined as the distance between two neighborhoods from the point of view of the residents of the first neighborhood. More specifically, we say that with respect to neighborhood A, neighborhood B is near, far, or remote. The degree to which A is affected by events in B tapers off with distance, and is zero if B is remote.

Proximity need not be symmetric. If neighborhood A contains popular destinations, it might be considered nearby by neighborhoods which its residents consider to be far away.

3.2.3 Neighborhood Effects Delay

Events and situations occurring in the Ground Area drive attitude change. Such drivers usually have a direct effect in a single neighborhood, but may have indirect effects in other neighborhoods. Athena 3 allows these indirect effects to be delayed by some number of days, to reflect the spread of the news across the playbox.

This portion of the model was inherited from JNEM, and makes good sense in a scenario where events are occurring minute by minute, hour by hour. In Athena, events take place day by day, and more usually week by week. The effects delay makes much less sense at this timescale, and in an era of modern communications and transport; consequently, it is likely to be removed in a future version.

3.2.4 Local vs. Non-Local Neighborhoods

Athena assumes that the neighborhoods that make up the playbox are more or less and contiguous and have a single more or less unified economy. Sometimes, however, it can be convenient to include neighborhoods in the scenario that are outside the economy. Pakistan is greatly affected by the decisions made by actors in India, for example; in a scenario involving the inner workings of Pakistan, it might be desirable to include India as a neighborhood while excluding India from the modeled economy. Thus, we can mark neighborhoods as *local* (participants in the local economy) or *non-local* (excluded from the local economy).

3.2.5 Production Capacity

The Economics model assigns production capacity (e.g., factories, farms, and other businesses) to neighborhoods at time 0 based on the size of the economy in dollars and the size of the labor force in each neighborhood. Each neighborhood is then assigned a production capacity factor (PCF) of 1.0 that reflects this initial production capacity. The PCF of a given neighborhood can be increased or decreased over time by actors or by the analyst to reflect construction of new facilities or damage to infrastructure, thus increasing or decreasing the production capacity of the economy as a whole.

Note that the costs associated with repairing, replacing, or building new production capacity are not modeled, nor is the nature of the required plant or the training of the labor force.

3.3 Actors

Actors are significant decision makers in the playbox. Depending on the needs of the scenario, an actor may be an individual, a committee, a small group, a ruling body, or an entire country. The essential point is that an actor has:

- Goals it wishes to achieve
- A strategy for achieving them
- Assets to use to achieve them.

In Athena 3, actors have income, which can be spent in a variety of ways, and can own force and organization groups which they can make use of.

Actors can have different domains—military, political, economic, cultural—and although these domains are implicit in the actor's assets and strategies rather than explicit in the model, they are no less important for all that.

Most actors participate in the political process, seeking control of neighborhoods or supporting those who do. Organization groups, which usually do not play politics in the playbox, are often owned by a "pseudo-actor", an actor that supports no one politically and exists only to hold the strategy that determines organization group behavior.

There is an additional pseudo-actor, or *agent*, the **SYSTEM** agent. The **SYSTEM** agent is not an actor, and cannot be involved in politics; however, it does have a strategy, which allows the analyst to set up various events and situations to occur when conditions in the simulation are right. In particular, the **SYSTEM** agent is responsible for displacing civilians to other neighborhoods.

3.4 Groups

Athena represents masses of human beings as *groups*. There are three kinds: *civilian groups*, *force groups*, and *organization groups*. Every group has a demeanor, **apathetic**, **average**, or **aggressive**, that indicates its propensity for violence.

3.4.1 Civilian Groups

The population of the playbox is broken down into a number of *civilian groups*. Each civilian group resides in a particular neighborhood, and has a *belief system* that gives it its identity (Section 5.1) and determines how it relates to other groups and actors. It also has greater or lesser *satisfaction* with the current state of affairs, and a greater or lesser *cooperation* with (willingness to give information to) the actors' various force groups (Sections 5.4 and 5.5). Civilian groups support actors to a greater or lesser degree; ultimately, any successful actor must either derive his power in a neighborhood from the resident civilians, or expect to keep a significant body of troops in place.

In JNEM, and in earlier versions of Athena, it has been usual to think of civilian groups in ethnic terms, as clans, tribes, or related groupings of such. At the beginning of the Athena project, in particular, we expected each group to be a stovepipe containing individuals from cradle to grave. In Athena 3, by contrast, groups are collections of people living in the same area who share a belief system. When an ethnic grouping contains significant divisions, e.g., a "generation gap", it may be prudent to split the ethnic grouping into multiple groups. Similarly, if a particular belief system cuts across ethnic groups, it may be wise to treat it as a single group.

At present, individuals cannot move from one group to another group. In future versions of Athena we might support this, as a way of indicating the erosion of one belief system in favor of another.

3.4.2 Force Groups

A *force group* is an organized group of personnel intended to project and use force in a neighborhood. Force groups belong to actors, and do their bidding according to the actor's strategies. In particular, an actor can increase his support in a neighborhood by moving his troops into that neighborhood. There are five kinds:

- Regular military
- Irregular military (e.g., militias)
- Paramilitary (e.g., SWAT teams, militarized police forces)
- Police
- Criminal

Each force type has a different force multiplier; regular military, for example, is much better at projecting and using force than normal police.

Force groups can carry out military duties of various kinds, and can also participate in humanitarian relief efforts (Sections 3.6 and 3.12.1).

Each force group starts the scenario with a certain number of troops, which can increase and decrease as the simulation runs. At present, force groups cannot recruit from the civilian population, though this is an area in which we are actively working.

Force groups do not have belief systems of their own; rather, they inherit them from their owning actors.

3.4.3 Organization Groups

Organization groups are similar to force groups, but have some mission other than the projection and use of force. There are three kinds:

- Non-Governmental Organizations (NGOs), e.g., Doctors Without Borders
- Inter-Governmental Organizations (IGOs), e.g., UNESCO
- Contractors, e.g., Halliburton

Like force groups, members of organization groups can be assigned activities in neighborhoods, though they are limited to humanitarian relief activities of various kinds. In principle organization groups can support the political goals of actors by their presence in one neighborhood or another; however, organization groups are usually owned by pseudo-actors who are not politically active.

3.5 Deployment

Every person in the playbox has to be somewhere, i.e., has to be located in some neighborhood. Civilians are simply located in their home neighborhoods (unless displaced; see Section 3.6); force and organization group personnel need to be *deployed* to particular neighborhoods.

As the simulation progresses, deployment is determined by the tactics chosen by the actors; see Section 6. It is also necessary to know where force and organization group personnel were located prior to the start of the simulation; this is called the *status quo deployment*, and is a scenario input.

Troops are deployed by their owning actors during strategy execution, and remain in place throughout the week until the next strategy execution.

3.6 Activity Assignment

Force and organization group personnel deployed to a neighborhood can be assigned activities in that neighborhood: patrolling, guarding, law enforcement, various kinds of humanitarian relief, and so forth. See Section 15.1 for a complete list of activities. These activities affect the attitudes of the civilian population. Activities are assigned by the actors during strategy execution, and like deployments take place over the following week.

Force and organization group activities have security requirements; a body of troops might be tasked to do humanitarian relief of some kind, but if they have insufficient security in the neighborhood their efforts will be of no avail (Section 3.8). See the *Athena Analyst's Guide* and the model parameter database (Section 12.16) for the security requirements.

Civilians are displaced by assigning them the **DISPLACED** or **IN_CAMPS** activities, which are assigned not by actors but by the System agent (Section 3.3) using the **DISPLACE** tactic.

3.7 Units

All deployed personnel, and all civilian personnel, are placed in *units*. The name derives from the classic military term; in Athena it simply means a collection of personnel belonging to the same group and assigned the same activity. Units have no distinctive or long-running identity; they are created during strategy execution and represent the location and activity of group personnel over the following week. If personnel from force group A are deployed to neighborhood B, and assigned various activities, then A will have at least one unit for each activity, plus an additional unit for those personnel that remain unassigned.

Units are useful for visualization; and many of the subsequent models in the Ground Area operate on units.

3.8 Volatility and Security

A neighborhood can be a safe or unsafe place to be for the people within it—and to a great extent, that depends on who they are and who is in the neighborhood with them. Athena computes two measures, the *volatility* of each neighborhood and the *security* of each group in each neighborhood.

Both depend on the personnel in the neighborhood and in nearby neighborhoods, and on their relationships with each other (see Section 5 for more on relationships).

First, each group in the neighborhood (whether civilian, force, or organization) can project a certain amount of force, given the kind of group it is and the number of personnel present. This is its power to defend itself. Civilian groups project minimal force per person, given that civilian groups include the very old and the very young, and many adults who are not inclined

to project force. Force groups exist to project force, and do it rather better; their effectiveness depends on the kind of force they are. Regular military projects the most force per person. Of organization groups, only contractors project any force (private security guards)—more than civilians but less than any force group. (See Section 3.4 for more on the different kinds of group.)

But a group's ability to defend itself does not depend solely on its own force—they may have friends to help them; and unless they have enemies they will not need to defend themselves. Athena totals up the force available to each group, including friends in the same neighborhood and (to a lesser degree) friends in nearby neighborhoods; and similarly it totals up the force available to the group's enemies in the same neighborhood and (to a lesser degree) in nearby neighborhoods.

The *volatility* of a neighborhood ranges from 0 to 100, and is a measure of how dangerous the neighborhood is to a random passerby given the degree of enmity present in the neighborhood, i.e., how likely a person is to get caught up in random violence that does not directly concern him.

Danger to a group comes from its enemies and from the kind of random violence measured by volatility. We capture this as the *security* of the group in the neighborhood. Security is an abstract measure ranging from -100 to 100. For actual use we convert it to a qualitative measure (high, medium, low, or none), or to a multiplicative factor using a Z-curve function.¹

In Athena 3, *volatility* is primarily a component of *security*, whereas *security* affects many things, including:

- Whether force and organization groups can carry out particular activities in a neighborhood.
- The degree to which groups can actively support the actors of their choice.

The addition of a military force to a neighborhood can greatly change the security of all groups in the neighborhood. In Athena 3, this change is due purely to the number of troops and the relationship between the force group and the other groups in the neighborhood: the security of the force group's enemies will decrease, and the security of the force group's friends will increase.

Consequently, further work needs to be done here; a force assigned peacekeeping duties should have a different effect than a group intending rapine and pillage.

¹ A piece-wise linear approximation to an S-curve. See the *Mars Analyst's Guide* for details.

3.9 Coverage

Coverage is a measure, from 0.0 to 1.0, of the fraction of a neighborhood or group affected by some situation. The notion of coverage is used in a number of places in Athena:

- Environmental situations are assigned a coverage when they are created. (Section 3.11)
- Coverage is computed for the mere presence of a military force deployed in a neighborhood.
- Coverage is computed for activities assigned to groups of all kinds. (Section 3.6)

For environmental situations, the coverage is simply an input. For presence and activities, it is a function defined by the number of troops required to achieve 2/3rds coverage given the size of the population. For presence, for example, presence coverage is 2/3rds when there are 25 troops present for each 1000 people in the civilian population. Coverage drops to zero when there are no troops present or engaged in the activity, and increases asymptotically to 1.0 as troops are added above the 2/3rds mark.

Assigned activities usually have a security requirement. If there are 25 troops per 1000 people assigned to do the **CMO_HEALTHCARE** activity, but the security of those troops is low, they cannot carry out the activity effectively and hence the coverage of that activity by those troops is 0.0.

The coverage fraction is used as a multiplier in the relevant rule set in the Driver Assessment Model (Section 5.7); and presence coverage is used in a variety of places, most notably in the Athena Attrition Model (Section 3.12).

3.10 Activity Situations

When a group is conducting an activity of a particular type in a neighborhood with coverage greater than 0.0, we have what we call an *activity situation*,² or “actsit”. Activity situations are created when coverage exceeds 0.0, and are destroyed when coverage returns to 0.0. So long as the situation persists it will have effects on civilian attitudes as determined by the relevant rule set in the Driver Assessment Model; see Section 5.7 and the *Athena Rules* document.

Force and organization group activities can mitigate particular environmental situations, e.g., the **CMO_HEALTHCARE** activity will have a greater effect on civilian attitudes when there is a **DISEASE** environmental situation in the neighborhood.

² For the purposes of this section, both mere presence and assigned activities count as activities.

3.11 Environmental Situations

Environmental situations, or “ensits,” represent problems in a neighborhood’s environment that adversely affect the resident civilians, e.g., power outages and food shortages; see Section 14.8 for the complete list.

Environmental situations are usually created by the analyst, or by actors using the **EXECUTIVE** tactic. An ensit will typically have a big negative effect on satisfaction on inception, a continuing negative effect so long as the situation persists, and a big positive effect when the situation is resolved.

The duration of an ensit can be set when the ensit is created; it can also be resolved explicitly by the analyst or by an actor using the **EXECUTIVE** tactic. Each ensit also has a coverage fraction, nominally 1.0, which can be decreased to decrease the ensit’s effects.

The effect of certain environmental situations can be mitigated by appropriate force and organization group activities, as indicated in the *Athena Rules* document.

The environmental situation model is one of the oldest parts of Athena, being adopted with minimal changes from JNEM. It was designed for five-day real-time training exercises with the intent of rewarding commanders for quick resolution and punish them for delayed or omitted resolution of the problems represented by the ensits. As such, it will tend to run “hot”; when ensits are used in an Athena scenario, the analyst should monitor them closely, and should consider using smaller coverage fractions to reduce the effects if they are overstated.

In the future, it is likely that many of the existing ensit types (e.g., power outages) will be replaced by service-oriented models like the current Essential Non-Infrastructure (ENI) Services model (Section 3.13), which is more suited to the Athena time frame.

3.12 Athena Attrition Model

Athena was designed to support Stability and Recovery Operations (S&RO); i.e., to model regions in which the heavy metal force-on-force battles are over (or have not yet begun). Thus, Athena 3 does not model full-on force-on-force attrition. Rather, it deals with two kinds of conflict: the efforts of conventional uniformed forces to hunt down and kill non-uniformed insurgent/terrorist forces, and the efforts of these non-uniformed insurgents and terrorists to use guerilla tactics against the uniformed forces. Such combat results in attrition to the relevant forces, thus reducing their numbers in the playbox, and also in civilian collateral damage with the relevant effects on civilian attitudes.

In short, uniformed forces can seek to attack non-uniformed forces, and non-uniformed forces can seek to attack uniformed forces, neighborhood by neighborhood.

3.12.1 Rules of Engagement

Whether force group A seeks to attack force group B in neighborhood N is determined by A's rules of engagement (ROE), which are set according to the strategy of the actor that owns group A. Using the **ATTROE** tactic, the actor can direct that A may attack B in neighborhood N up to some number of times over the next week. If A is a non-uniformed group, then the actor may also specify whether A is to minimize its own losses or maximize damage to B.

In the current model, civilian collateral damage occurs when a uniformed force attacks a non-uniformed force, and when a uniformed force defends itself against attack by a non-uniformed force. Thus, uniformed forces also have a defending ROE in each neighborhood, which determined whether and how quickly they fire back at attacking non-uniformed forces. This directly affects the quantity of civilian casualties.

3.12.2 Presence and Intelligence

Just because force group A has been directed to attack force group B in neighborhood N, it is not certain that it will be able to. Whether attacks occur or not depend on a number of circumstances:

- Both A and B must have troops in neighborhood N.
- The more troops A has, the more likely it is to be able to find and attack B.
- The more troops B has, the more easy it is to find.
- Intelligence, as indicated by the cooperation of the neighborhood with both groups, also plays a role.
 - If A gets better cooperation than B, it will have an easier time finding and attack B.
 - If A gets worse cooperation, then it will have a harder time.
 - If A is a non-uniformed force, then the expected losses must be acceptable, and this also depends on the quality of the intelligence received by A, as indicated by the neighborhood cooperation.

Cooperation is discussed in Section 5.5.

3.12.3 Attrition Assessment

The number of successful attacks by all parties, and the resulting civilian casualties, are assessed at the end of each week just prior to the next strategy execution. The casualties are then given to the Driver Assessment Model (Section 5.7) so that the attitude changes can be assessed.

3.12.4 Magic Attrition

The Athena Attrition Model does not address terror bombings, assassinations of political figures, or deaths due to other kinds of armed combat than those described above. And yet, these kinds of deaths occur. For this reason Athena provides the ability to do “magic attrition,” which can be initiated by the analyst, or by an actor or the **SYSTEM** agent using the **EXECUTIVE** tactic. Magic attrition can affect members of any group; and in particular, civilian casualties will be assessed by the Driver Assessment Model just like casualties resulting from the kinds of combat Athena *does* model.

Note that magic attrition should not be used for civilian deaths due to natural disasters, epidemics, or other causes that do not involve combat. For those kinds of things, the attitude effects should be handled by either environmental situations (Section 3.11) or magic attitude drivers (Section 5.8).

3.13 Essential Non-Infrastructure (ENI) Services

Essential Non-Infrastructure (ENI) Services are services provided to civilians in a neighborhood by an actor, the absence of which causes hardship but which do not require substantial infrastructure to provide. Provision of services is controlled by the actor's strategy, and can be targeted to specific groups in the neighborhood, ignoring others.

3.13.1 The Notion of a Service

A service is something provided to the civilians (possibly by their own efforts, as enabled or supported by actors) that has a level that can increase or decrease over time. Examples are power service, postal service, communications, water supply, the court system and other governmental services, and the like (though not law enforcement, as that's an assigned activity). We call the level of service for a particular service the *LOS*. For any given service there are four specific levels of service that are of interest:

The Actual Level of Service (ALOS)

How much of the service is the group actually receiving at the present time?

The Required Level of Service (RLOS)

How much of the service does the group need to live without significant hardship?

The Expected Level of Service (ELOS)

How much of the service is the group accustomed to getting?

The Saturation Level of Service (SLOS)

What's the level of the service which saturates the demand? Once the civilians have all they want, they don't care if more is available.

The units appropriate for measuring a particular the level of a particular service will vary from service to service.

The expected level of service will slowly approach the actual level of service over time; in other words, the civilians will eventually become accustomed to whatever level of service they receive. Expectations will rise more quickly than they will fall: we become accustomed to good things more quickly than we become resigned to bad things.

For example, in most of America ELOS for the power supply simply *is* the SLOS. We most of us have all the power we are willing to buy. If the power is out, we are immediately unhappy, and would takes us quite a while to get used even to power provided on a regular if intermittent schedule; but when the power goes back on, we get used to it with great rapidity.

Civilian attitudes improve when the ALOS is greater than expected (though not more than the SLOS), and worsen when the ALOS is less than expected, and especially if it is less than required.

There are four cases of particular interest:

- Case R-: Service is less than required
- Case E-: Service is less than expected
- Case E: Service meets expectations
- Case E+: Service is better than expected

Note that case R- trumps all of the others; and that case E+ can only occur if the expected level of service is less than saturation.

At the present time we have used this paradigm only for ENI services; we expect to make use of it for infrastructure-based services in the future.

3.13.2 Measurement of ENI Services

Actors provide ENI services to groups in neighborhoods by spending money on them using the **FUNDENI** tactic. No infrastructure is required, by definition; and we assume that every dollar spent translates (not necessarily linearly) into service provided.

For convenience, we measure the provision of ENI services to a group in a neighborhood as a fraction of the saturation level of service (SLOS) for that group: 0.0 implies no service, and 1.0 implies the saturation level of service. Then, we specify the saturation level of service by the per capita funding required to achieve it. If actors provide funding for more than the saturation level of service, the ALOS will be greater than 1.0.

3.13.3 Required Level of ENI Services

The required level of ENI services is set in the model parameter database as a fraction of the saturation level of service, according to the urbanization level of the neighborhood.

3.13.4 Status Quo ENI Funding

The level of ENI service provided to a particular group depends on the actor's strategies; but how the actors execute their strategies depends on the current state of affairs, which includes some level of spending on ENI Services. Thus, we need to know the funding for ENI services prior to time 0; this is a scenario input called the *status quo ENI funding*.

3.13.5 Effects of ENI Services

The current level of ENI services affects two things:

- Civilian satisfaction levels; see Section 5.7.
- The vertical relationships of civilian groups with actors; see Section 5.3.

In each case, the fundamental questions are whether the civilians are receiving the required level of service; and if so, whether they are receiving more or less service than they expect.

In terms of the vertical relationships, it also matters whether or not the actor providing the service has control of the group's neighborhood; see Section 5.3 (and the *Athena Analyst's Guide*) for details.

3.13.6 Services vs. Environmental Situations

As such, the service paradigm is an improvement over the Environmental Situation paradigm for services like the power system and the water supply. Using the power system for illustrative purposes, the ensit paradigm implicitly assumes that the ALOS is normally at its expected value, and that when problems occur it drops down to 0.0. Horrors ensue until the problem is resolved, at which point it service returns to its previously expected level.

In a long-run scenario, however, it is quite possible that the power service may be substandard (though not zero) for quite long periods of time. Power for 12 hours a day is much better than no power at all; and after a few weeks' time, the civilians will begin to adjust to it (and the attitude effects will cease). If power then drops to 4 hours a day, they will again react negatively; but if it returns to 24 hours a day they will react positively.

We expect service-orient models to replace many of the existing ensit types as time goes on.

4. DEMOGRAPHICS

The Demographics Area is closely tied to the Ground Area, as it is concerned with how many civilians there are and where they live. The Demographics model proper is responsible for determining the current population (by group, neighborhood, and playbox), as well as the size of the labor force, the number of consumers in the local economy, and similar population statistics. In addition, the Demographic Situation (demsit) model determines the effects of unemployment on each civilian group, which in turn drives attitude change.

4.1 Base Population

The population is divided into civilian groups (Section 3.4.1); each civilian group resides in a neighborhood. At time 0, each civilian group has an initial or *base* population. The base population of the neighborhood is simply the total across the civilian groups, and the base population of the playbox is simply the total across the neighborhoods. Note that we also track the total population of local neighborhoods (Section 3.2.4), because that figures into the Economics Area.

4.2 Current Population

The *current population* of civilian groups and of neighborhoods can change over time. Athena does not model births or natural deaths, but it does model deaths due to civilian collateral damage (Section 3.12). The Demographics model tracks attrition to date, subtracting it from the current population. In addition, civilian group personnel can be *displaced* to other neighborhoods (Section 3.10), which removes them from the current population of their group and neighborhood, and adds them to the current population of the neighborhood to which they are displaced.

4.3 Subsistence Agriculture

Civilian personnel can support themselves by *subsistence agriculture* or by participating in the local economy. In Athena 3 there is a hard line between the two: any given person is in one subset or the other. The percentage of each civilian group that lives by subsistence agriculture is a scenario input.

When civilians belonging to a group are displaced from their home neighborhood, it is presumed that subsistence and non-subsistence personnel are displaced in proportion to the

size of the two subsets within that group. If they later return to their homes, they resume their previous ways of life.³

Because subsistence personnel do not (by definition) participate in the local economy, they are neither consumers nor members of the labor force. As a result, they are not directly affected by high unemployment rates.

4.4 Consumers and Workers

All non-subsistence-agriculture personnel are presumed to be *consumers* in the local economy;⁴ the total number of consumers drives the size of the economy.

In each group, only a percentage of the consumers (nominally 60%) are members of the *labor force*. For civilians displaced from their homes, whether they remain within their own neighborhood or are displaced to another neighborhood, the percentage drops to 40%. Displaced civilians who are settled in camps do not contribute to the labor force at all.

4.5 Demographic Situations

Demographic situations are situations detected by the Demographic model that affect the attitudes of the civilians. Athena 3 defines only one demographic situation, or “demsit”, the Unemployment situation.

The unemployment rate is computed for the entire playbox by the Economics model. It affects civilian groups in proportion to the number of workers in each group. Given that, high unemployment affects civilians in two ways:

- Directly, by economic hardship to members of the group. Groups with a high Subsistence Agriculture Percentage are relatively immune to this.
- Indirectly, by the presence of numbers of unemployed workers in the neighborhood.

See the *Athena Rules* document for specifics.

³ This is not realistic; displaced subsistence farmers usually lose their land and livestock, and hence cannot easily go back to subsistence agriculture.

⁴ Except, of course, in non-local neighborhoods. The number of consumers and laborers in these neighborhoods is ignored.

5. ATTITUDES

The *Attitudes Model* deals with the attitudes of the people in the playbox, and particularly:

- The belief systems of each of the actors and civilian groups
- The horizontal relationships between groups
- The vertical relationships between groups and actors
- The satisfaction of the civilian groups with respect to various concerns
- The cooperation (i.e., willingness to share information) of the civilian groups with respect to the force groups.
- Assessment of the effects of events and situations in the other models on the attitudes.

Note that the term “attitudes” properly applies to satisfaction and cooperation levels, which are managed by the Generalized Regional Attitudes Model (GRAM) (see the *Mars Analyst's Guide*). However, the relationships (which derive from the belief systems, among other things) are also attitudes in a wider sense.

5.1 Belief Systems and Affinities

Every civilian group and actor has a *belief system*, a statement of the ideas and issues that are important to the group or actor, along with how important they are and how the group or actor feels about those who disagree. For a civilian group, the group's belief system is the source of the group's identity and the basis for its relations with all other groups. An actor's belief system may indeed reflect the actor's deeply held beliefs, or it may be a construct intended to garner support from the civilians.

Given the belief systems of two entities, A and B, we compute the *affinity* of A with B, and of B with A. The affinity is a number from 1.0 down to -1.0 that indicates whether A supports or opposes the same things as B. Note that affinity need not be symmetric.

All horizontal and vertical relationships in Athena are ultimately based on affinities, and hence on belief systems.

Belief systems and affinities are computed by the Mars Affinity Model (MAM), which is documented in the *Mars Analyst's Guide*.

5.1.1 Beliefs and Topics

A belief system consists of an entity's beliefs about one or more topics. A topic is some value, principle, or issue about which there is some disagreement in the playbox. In Pakistan, for example, Islam is a significant fault line between the Pakistani citizens and the United States. Topics are chosen by the analyst; there is no default set.

A belief consists of two values, the entity's *position* for or against the topic of interest, sometimes known as the *zeal*, and the entity's *emphasis* on agreement or disagreement with that position. The former indicates how much the entity cares, and the latter determines how it feels about those who agree or disagree.

The position and emphasis are usually entered qualitatively, using the following values. For position, the entity may be:

- **Passionately For (P+)**
- **Strongly For (S+)**
- **Weakly For (W+)**
- **Ambivalent (A)**
- **Weakly Against (W-)**
- **Strongly Against (S-)**
- **Passionately Against (P-)**

The entity may put its emphasis on agreement or disagreement, as follows:

- **Agreement—Strong**
- **Agreement**
- **Neither**
- **Disagreement**
- **Disagreement—Strong**
- **Disagreement—Extreme**

If the emphasis is on agreement, the entity will tend to have a higher affinity with those entities with whom it agrees on this topic, while to some extent disregarding disagreements. If the emphasis is on disagreement, the entity will tend to have a lower affinity with those with whom it disagrees on this topic, while to some extent disregarding agreements.

Note that strong beliefs do not thereby imply a propensity to use violent action (though they may engender violent action in those groups that are so inclined). Propensity to violence is driven by a group's *demeanor*; see Section 3.4. For example, I might be passionately in favor of chocolate ice cream, and greatly dislike anyone who passionately dislikes chocolate ice cream without being inclined to take violent action against the chocolate ice cream haters. It depends on my demeanor.

Levity aside, a strong position and emphasis on a topic *does* indicate some willingness to take action. It simply does not indicate a propensity for violence.

5.1.2 Affinity

The affinity between two entities is computed by comparing their beliefs on each topic, and tallying the effects of their agreements and disagreements given their positions and emphases. As stated above, affinity is a number from -1.0 to 1.0 that is used as the basis for computing relationships.

Affinity is computed when the scenario is locked and simulation begins; in Athena 3 it is constant thereafter. In future versions we expect to allow it to vary.

5.1.3 Playbox Commonality

Playbox commonality is the degree of belief that is generally common to all groups in the playbox, to the degree that they are part of the dominant culture.

Selecting the relevant set of topics for a given playbox is more of an art than a science, and by the nature of things the tendency is to accentuate the negative—it is simply more easy to identify fault lines rather than the significant areas of agreement. When commonality is ignored, however, the resulting affinities tend to indicate that all parties concerned hate each other with a deep and abiding hatred.

The Mars Affinity Model handles this using the *playbox commonality* slider, a numeric factor nominally set to 1.0. We assume that the entities in the playbox have significant commonality, and in fact generally agree on about as many things as they disagree. More specifically, for each topic entered by the analyst, we add an implicit pseudo-topic of general agreement.

The number of these pseudo-topics is determined by the slider. If it is set to 1.0, there are as many pseudo-topics as real topics. If it is set to 2.0, there are twice as many; if it is set to 0.5, there are half as many. Moving the slider up and down will tend to move affinities up and down accordingly.

5.1.4 Entity Commonality

The *entity commonality* is the extent to which the entity participates in the general consensus indicated by the playbox commonality.

The playbox commonality lets you establish the degree of belief that is generally common to all groups in the playbox. But in fact, different entities will share in that common pool of belief to different extents, and foreign groups and actors will often have significantly different beliefs. Consequently, each group and actor has an *entity commonality* slider: a number from 0.0 to 1.0. If the entity commonality is 1.0, the entity shares the general consensus completely; if 0.0, not at all.

Decreasing an entity's entity commonality will tend to decrease the entity's affinity with other entities.

5.2 Horizontal Relationships

A horizontal relationship is a representation of how a group feels about another group. Every group, whether civilian, force, or organization, has a horizontal relationship with every other group. The relationship is represented as a number for -1.0 to 1.0, where 1.0 indicates that the groups are bosom friends (every group automatically has a relationship of 1.0 with itself) and -1.0 indicates that the groups are the bitterest of enemies. The realistic range is from about -0.8 to about +0.8.

In Athena 3, these relationships are simply equal to the relevant affinities, and are constant for the duration of the simulation. Force and organization groups do not have belief systems, but they are owned by actors that do; and for the purpose of computing horizontal relationships we simply presume that they inherit the belief systems of their owners.

This horizontal relationship is one of the most basic concepts in Athena, and has significant effects across the entire simulation.

5.2.1 Relationship Overrides

Every computed horizontal relationship can be overridden by an analyst's preferred value during scenario preparation.

5.3 Vertical Relationships

Just as each group has a horizontal relationship with every other group, it also has a vertical relationships with each actor, that represents how it feels about that actor; this is the basis for the actor's support and influence. Unlike horizontal relationships, which are bidirectional (though asymmetric), vertical relationships are unidirectional—that is, we measure each group's relationship to each actor, but not the actor's relationship to each group. Actors are what they do, and what they do is determined by their strategies, not by their affinities.

Like horizontal relationships, vertical relationships are measured from -1.0 to 1.0, a range which is often expressed qualitatively:

- Supports
- Likes
- Is Indifferent To
- Dislikes
- Opposes

Vertical relationships play a major role in the political model, for they are the basis for computing each actor's support and influence and for the determination of neighborhood control (Section 6.2).

5.3.1 Force and Organization Groups

Every actor has an affinity for every other actor, and every force and organization group is owned by an actor. A force or organization group's vertical relationship with its owner is always 1.0, and its vertical relationship with any other actor is simply its owner's affinity for that actor.

5.3.2 Civilian Groups

The vertical relationship of a civilian group with an actor is rather more complicated, as it can vary dynamically as the simulation runs. It is based on the affinity of the group for the actor, but this is adjusted by a number of factors:

- Whether the actor is or is not in control of the group's neighborhood
- How the group's mood (see Section 5.4) has changed
- Whether and to what extent the actor is providing Essential Non-Infrastructure (ENI) services to the group

The baseline for assessing these factors is the start of the simulation; or, later on, the time at which control of the neighborhood last shifted—an actor newly in control is judged on the state of affairs on his watch, rather than his predecessor's.

Future versions of Athena will probably take additional factors into account:

- Tactics chosen by the actor, and how they accord with the group's belief system
- Changes in the group's or actor's belief systems, resulting in a change in affinity.

5.4 Satisfaction Levels

Every civilian group has a sense of satisfaction or dissatisfaction with the state of affairs in the playbox. Satisfaction in this sense is not a feeling, *per se*, though we often use the language of feelings and take about the group's "mood" or say that the group "likes" or "dislikes" some event or situation. Rather, dissatisfaction is the will to change the current state of affairs, and satisfaction is the will to preserve the current state of affairs.

Satisfaction is measured as a number from 100.0 to -100.0, where 100.0 is perfectly satisfied and -100.0 is utterly dissatisfied.

The initial satisfaction levels are set by the analyst during scenario preparation; once simulation begins they vary depending on the events and situations that occur (Section 5.7).

5.4.1 The Four Concerns

We measure satisfaction on four axes, called the four *concerns*, also known as the four *soft factors*:⁵

Autonomy (AUT): Is the group satisfied with their government? AUT measures how a group feels about its governance. This includes how it feels about the actors that govern it, its ability to participate in the political system, and whether or not it has its preferred system of governance.

Safety (SFT): Do members of the group fear for their lives, either from hostile attack or from collateral damage from force activities? This fear includes environmental concerns such as life-threatening disease, starvation, and dying of thirst.

Culture (CUL): Does the group feel that its culture and religion, including cultural and religious sites and artifacts, are respected or denigrated?

Quality of Life (QOL): Is the group satisfied with their surroundings? QOL measures how a group feels about its prosperity, living standards, and economy. This includes the physical plants that provide services, including water, power, public transportation, commercial markets, hospitals, etc., and those things associated with these services, such as sanitation, health, education, employment, food, clothing, and shelter.

5.4.2 Saliencies

Each group has a *saliency* for each concern: a number from 1.0 to 0.0 that indicates how important that concern is to the group, where 1.0 is crucial and 0.0 is negligible.

5.4.3 Group Mood

A group's *mood* is the average of its four satisfaction levels, weighted by the group's saliency for each concern. Mood is a convenient summary statistic, and is an input to a number of other models.

⁵ The Athena team invariably uses the term "concern"; our sponsors seem to prefer the term "soft factor".

5.5 Cooperation Levels

Cooperation is the measure of how much information a civilian group, whether willing or coerced, will provide to a force group. Each civilian group is said to have a *cooperation level* with respect to each force group. The cooperation level is a number from 0 to 100, and represents the probability that a member of the civilian group will give information to a member of the force group.

Like satisfaction levels, cooperation levels are initialized during scenario preparation and vary thereafter based on the events and situations that occur (Section 5.7).

Note that having a high cooperation does not imply that the civilian group will overtly aid the force group in any way—they might or might not.

5.6 Trends and Thresholds

Each satisfaction and cooperation curve has associated with it an ascending trend and threshold, and a descending trend and threshold; these allow the curve's value to regress to a mean value, or at least a mean range. A trend is simply a slope effect (Section 5.7.1) that is applied to the curve each day; the ascending trend increases the level, and the descending trend decreases it. Each is associated with a threshold. If the current level is above the ascending threshold, the ascending trend is ignored; if the current level is below the descending trend, the descending threshold is ignored.

For example, it might be reasonable to assume that a group's satisfaction with Quality of Life (QOL) is 0.0 in the long term: the civilians get used to current conditions and don't think about them one way or another. This can be modeled by setting an ascending trend with an ascending threshold at or just below 0.0, and a descending trend with a descending threshold at our just above zero. The ascending trend will then drive the current QOL level up to 0.0 and stop, and the descending trend will drive the level down to 0.0 and then stop.

All trends default to 0.0, and thus have no effect.

5.7 The Driver Assessment Model (DAM)

The *Driver Assessment Model* (DAM) is responsible for assessing the satisfaction and cooperation effects of the various events and situations that occur in the simulation, e.g., civilian casualties (Section 3.12.3) and activity situations (Section 3.10). Collectively, these events and situations are called *drivers*. Athena contains a rule set for each kind of driver; the rule set assesses the driver, and gives *level inputs* and *slope inputs* (see below) to GRAM, which tracks satisfaction and cooperation effects as they play out over time. These rule sets are documented in detail in the *Athena Rules* document.

5.7.1 Level and Slope Inputs

A *level input* is a step change that takes effect over a short period of time, usually two days. A *slope input* is a slope (e.g., 5 points per day) that takes effect each day for the duration of a situation. Level inputs are usually used to model the effect of significant one-time events, and slope inputs are usually used to model the effect of on-going situations.

Both kinds of input have a *magnitude* which is applied to a particular curve.

5.7.2 Direct and Indirect Effects

Every level or slope input targets a specific curve: group A's satisfaction with QOL, or group B's cooperation with group C. This is called the direct effect. But other groups are affected as well. If group A takes casualties, for example, the satisfaction of the other groups in the neighborhood will be affected as well. This is called an indirect effect, and it is usually modified by the relationship between the groups.

Indirect effects can occur in the same neighborhood, in nearby neighborhoods, and in far-away neighborhoods (see Section 3.2.2 for a discussion of neighborhood proximity). The spread of indirect effects is determined by the input's *here factor*, *near factor*, and *far factor*, which is a simple multiplier of 0.0 to 1.0 applied to the magnitude of the indirect effect in the same neighborhood (here factor), in nearby neighborhoods (near factor) and in far-away neighborhoods (far factor). These factors are also commonly referred to as *s*, *p*, and *q*, respectively. The here factor, *s*, is 1.0 in all of the built-in rule sets, but may be set to other values by magic attitude drivers (Section 5.8). The near and far factors, *p* and *q*, are set in the model parameter database for all of the built-in rule sets; see the *Athena Rules* document for the default values, or query the model parameter database in the Athena application (Section 12.16).

5.8 Magic Attitude Drivers

The Driver Assessment Model's rule sets cover a wide variety of events and situations, but they don't cover everything; and in particular, they don't cover one-of-a-kind events like the assassination of a government leader or the World Trade Center attack. *Magic Attitude Drivers* (MADs) allow the analyst to create rule sets "on the fly". MADs can be created interactively by the analyst while the simulation is paused; alternatively, they can be created by an **EXECUTIVE** tactic. One way or another, they can affect both satisfaction and cooperation levels with both slope and level effects. See Section 11.11 for the user interface.

6. POLITICS

The Politics Area deals with actors and their strategies (goals, tactics, and attached conditions) along with the determination of support, influence, and neighborhood control. As described above in Section 2, the interplay of actor's strategies being executed over time is the engine that makes Athena run.

6.1 Strategies: Goals, Tactics, and Conditions

As described in Section 3.3, the actors are the significant decision makers in the playbox. Actors have *goals* that they wish to achieve, and *assets* to use to achieve them; they execute *tactics* using their assets to achieve these goals; and they decide which tactics to use, and whether or not their goals have been achieved, through the use of *conditions*.

This section will define all of these terms, and how they relate to each other. There are many different kinds of tactic, and many different kinds of condition; these are documented in the reference portion of this user's guide, in Sections 15 and 16, and in the on-line help.

6.1.1 Assets

Actors can have two kinds of assets: cash, and personnel.

An actor's personnel are simply the members of the force and organization groups owned by the actor; these groups are assigned to actors as part of scenario preparation. By executing the appropriate tactics, described in Section 15, the actor can mobilize, demobilize, deploy, and assign activities to personnel, as described in Sections 3.5 and 3.6.

With regard to cash, each actor has two pots of money, his *cash-on-hand*, money that is available to be spent, and his *cash-reserve*, money that is being saved for use at a later time. Tactics that cost money are paid for from the *cash-on-hand*; left-over cash rolls over to the next week. The actor can use tactics to move money between his *cash-on-hand* and his *cash-reserve*.

In addition, the actor has a weekly income. In Athena 3, this income is specified as part of the scenario; in future versions, we expect it to flow to the actor from the Economic model. The income flows into his *cash-on-hand* at the beginning of strategy execution each week.

Finally, the actor can be funded by other actors. When actor A funds actor B, he does so by executing the **FUND** tactic, which transfers some quantity of A's *cash-on-hand* to B. The money becomes available to B the following week.

6.1.2 Conditions

Actors are decision makers, and so it is necessary to allow the analyst to express in the simulation the *conditions* which lead the actors to make one decision or another. A condition is a Boolean predicate, true if some particular state of affairs exists in the simulation, and false otherwise. Athena defines a variety of types of condition; for example,

- Does actor A control neighborhood N?
- Is actor A's *cash-reserve* greater than \$1,000,000?
- Is group G's mood less than -40.0?

The complete list is documented in Section 16, and in the on-line help. In addition, it is possible for the analyst to define complex conditions using Boolean expressions via the **EXPR** condition.

An actor's goals are defined in terms of conditions; and conditions can be attached to the actor's tactics, to determine when and if the tactics will be considered for use.

6.1.3 Goals

Actors have goals they would like to achieve. In Athena, an actor can have any number of goals; each consists of a set of one or more conditions, specified by the analyst. If the conditions are all true, the goal is said to be *met*, and if any condition is false the goal is set to be *unmet*.

Goals are attached to tactics in the form of the **MET** and **UNMET** conditions. A goal can represent either a state-of-affairs that the actor would like to bring about, in which case the actor will execute specific tactics if the goal is unmet, or a state-of-affairs that the actor would like to preserve, in which case the actor will execute specific tactics if the goal is met.

6.1.4 Tactics

A *tactic* is an action that the actor can choose to take, possibly in support of one or more goals. He may deploy troops to neighborhoods, assign troops to do particular activities, set rules-of-engagement, fund essential services, support other actors, and so forth. There are many different kinds of tactic; each kind is defined in detail in Section 15.

A tactic may have a cost in dollars, personnel, or both, depending on the tactic type and its parameters. Dollars spent on a tactic are consumed. Personnel used by a tactic are unavailable for use by other tactics during the same week. If the required assets are not available, the tactic cannot be executed.

Conditions may be attached to tactics; the tactic will only be considered for execution if all attached conditions are true. Thus, for example, a tactic can therefore be executed in support of a goal or goals by attaching the **MET** condition, or during a particular time interval by attaching the **DURING** condition.

Note that it is quite possible for an actor to take actions that are counter-productive given his stated goals. The Political model is not intended to compute optimal courses-of-action for the actors; rather, it is intended to allow the analyst to model the decision makers in the playbox, along with their limitations and prejudices, and track the consequences of the actions they actually take.

6.1.5 Strategy Execution

At the beginning of each week, every actor executes his strategy for the following week given the current state of affairs. This process is called strategy execution, and it is fairly straightforward:

- Each actor's tactics are stored in priority order; the order is determined by the analyst when the tactics are created.
- All goals are evaluated, given the state of affairs that exists prior to the start of strategy execution.
- The actor considers each of his tactics in order from highest priority to lowest priority.
 - If any of the tactic's conditions are false, the tactic will not be executed. This includes any goals attached via the **MET** or **UNMET** conditions.
 - If the actor has insufficient assets remaining, the tactic will not be executed.
 - Otherwise, the tactic is executed.

Note that "executing the tactic" does not mean that all of the tactic's work is done immediately. Many tactics set up a state of affairs, which then plays out in the simulation over the following week. When the actor executes a tactic, one can think of that as the actor giving the relevant orders to his subordinates.

6.1.6 What the Actor Knows, and When He Knows It

Athena assumes that all of the actors execute their strategies simultaneously, without any collusion or comparing of notes. Thus, as the actor works through the process of executing his strategy, he knows two things:

- The state of affairs before he started.
- The decisions he has already made during the current execution process.

For example, he knows how many troops he has already deployed during the current strategy execution process.

The relevant conditions are defined according to this scheme, following the basic rule: when actor A is querying his own assets and decisions, he sees their current values as of that point in strategy execution, but when he is querying the assets of and decisions of others he sees the information as of the beginning of strategy execution.

6.2 Support, Influence, and Control

We say that an actor “controls” a neighborhood when the residents of the neighborhood hold him responsible for dealing with the neighborhood’s problems. The actor in control is blamed when things go poorly, and praised when things go well; these things affect the vertical relationships between the residents and the actor (Section 5.3). The second significant role of the political model is to determine who is in control, and when control has shifted from one actor to another.

The basic paradigm is as follows: actors receive *support* from the people in the neighborhood (including force group personnel). An actor with sufficient support is said to have *influence* in the neighborhood. Depending on the relative influence values, the actor with the most influence is usually the one said to be “in control”.

6.2.1 Support

Group G is said to support actor A directly in neighborhood N if:

- G’s vertical relationship with A is strong enough (nominally, greater than 0.2)
- G’s security in N is high enough.⁶

In essence, group G has to *want* to support A, and has to have sufficient freedom of movement to be *able* to support A. Support is 0 if G does not support A, and increases from there with increases in population, vertical relationship, and security.

Actor A’s direct support in N is the sum of his support from all groups in N; the result is a number from 0.0 to 1.0, and can be thought of as the fraction of the neighborhood that supports A. Note that support is not a zero-sum game: group G can support any number of actors to differing degrees.

⁶ We use a Z-curve to convert the base security level, -100 to +100, into a multiplier ranging from 0.0 to (usually) 1.0.

Actor A can use this support in three ways:

- Actor A can try to gain influence in N
- Actor A can support some other actor in N
- Actor A can choose not to use his support at all, thereby bowing out of the political process all together. (Actors that own organization groups often do this.)

Support received from other actors is called derived support; and actor A's total support is simply the sum of the two. Note that because of derived support, an actor can find himself with enough influence to be in control of the neighborhood even if he gives his own direct support to another actor or to no one.

The analyst specifies, as part of the scenario, each actor's choice of whom to support by default: himself, some other particular actor, or no one. Then, the actor may change who he supports in each neighborhood from week to week using the **SUPPORT** tactic.

6.2.2 Influence

Actors have influence in a neighborhood in proportion to their support relative to other actors. If only one actor has support in the neighborhood, only that actor can have any influence. Note that influence requires at least a minimal amount of support, nominally 0.1; this prevents a negligible force (e.g., a Red Cross team) from moving into a neighborhood that's in a state of chaos and finding themselves in control despite having almost no support.

Influence is computed by taking the set of actors who meet the minimum support requirement, and normalizing their total support figures. Thus, an actor's influence is a number between 0.0 and 1.0; and influence *is* a zero-sum game. An actor cannot increase his own influence without decreasing the influence of the other actors.

6.2.3 Control

As part of scenario preparation, the analyst specifies which actor is in control of each neighborhood; this includes the possibility that *no* actor is in control of the neighborhood at the start of the scenario. Then, at the end of each week Athena looks to see which actor (if any) will be in control for the following week.

Suppose that actor A is in control of neighborhood N:

- If actor A has more than the threshold amount of influence (nominally 0.5) in neighborhood N, then actor A remains in control of N.
- If A has less than the threshold amount, but more than any other actor, then A remains in control, though in a rather precarious position.

- If some actor B has more than the threshold amount of influence, then B is now in control; **control has shifted**.
- If at least one actor has more influence than A but no actor has more than the threshold amount, then no actor has control of the neighborhood; **control has shifted**.

If no actor was in control of N, the rules are simpler:

- If some actor B has more than the threshold amount of influence, then B is now in control; **control has shifted**.

To put it in another words, the actor in control must be dominated by some other actor to be seen to have lost control; but that other actor might not be powerful enough to take control himself. Control of the neighborhood is then up for grabs.

6.2.4 When Control Shifts

When control does shift to a different actor, or no actor, the civilians respond to the change:

- There is a new political situation, and a new baseline is established for computation of vertical relationships (Section 5.3)
- The change affects the attitudes of the civilian groups resident in the neighborhood; see the **CONTROL** rule set in the *Athena Rules* document.

7. ECONOMICS

The Athena Economics model models the economy of the region of interest, which can be an entire country, a portion of a country, or several small countries taken together. We refer to this as the *local economy*. The core of the model is a three-sector Computable General Equilibrium (CGE) model, a system of non-linear equations solved using the Gauss-Seidel algorithm. The CGE is described in full detail in the *Athena Analyst's Guide*; this section will give an overview, and discuss how the Economic model relates to the rest of Athena.

Note: *The three-sector CGE is essentially a prototype and a proof-of-concept; while it is not without value, it is limited in its applicability. We have designed a more robust six-sector CGE, which we expect to implement in Athena 4. For this reason, users might wish to disable the Economic model; see Section 13.1 for instructions.*

7.1 Dollars

Athena represents money in “dollars”; however, no attempt is made to tie the value of an Athena “dollar” to that of a real dollar. If the inputs to the CGE are given such that Athena’s monetary units correspond to some real currency, then they do; and if not, then not.

7.2 Sectors

The CGE partitions the local economy into three sectors: **goods**, **pop**, and **else**:

The “goods” sector

The **goods** sector includes all production of goods and services in the local economy. The unit of production is the *goods basket*, abbreviated “GBasket”, a notional basket of goods and services nominally costing about \$1.

The “pop” sector

The **pop** sector includes all labor by the workers in the local region, and all consumption by the population of the local region. The unit of production is the work-year of an *average worker*. Just as the goods basket represents a notional bundle of goods and services, the average worker represents a notional bundle of skills and kinds of work.

The “else” sector

The **else** sector represents everything else in the local economy that isn’t covered by the first two sectors, as well as the entire rest of the world. In practice, the “rest of the world” means imports and exports. The unit of production is the *else basket*, abbreviated “EBasket”, which is similar to the goods basket and also nominally costs about \$1.

Each sector produces a certain amount of output; measured in dollars, this output is called the *revenue* of the sector. In the process of producing its output, the sector consumes the output of all three sectors; measured in dollars, this vector is called the sector’s *expenses*.

7.3 Shape vs. Size

We distinguish between the *size* of the economy, which can be roughly thought of as total revenues, and the *shape* of the economy, or the proportion of revenues across the sectors. The size of the economy is driven by consumption: increase the number of consumers, or the amount consumed by each, and the economy must increase in size. Decrease the amount of consumption, and the economy must shrink. But as the economy increases and decreases in size, its basic shape remains the same, because the basic industries and technologies in use remain the same.

The size of the economy is therefore driven by the number of consumers, which comes from the Demographic model (Section 4). The shape is determined by the following inputs:

- The base wage: the average wage for one work-year, in dollars.
- The average consumption of goods by each consumer each year, in goods baskets.
- The Cobb-Douglas parameters for the three sectors; these parameters determine the mix of inputs required to produce one unit of the sector's output, assuming a Cobb-Douglas production function.

Setting these parameters is called *calibrating the Economic model*; the process is described in Section 13.2.

In theory we should also set the base prices for the **goods** and **else** sectors, but as we have defined the goods basket and the else basket as baskets worth \$1 at time 0, the base price is naturally \$1.

7.4 Economic Outputs

The Economic model is run once a week. Each week, it produces the following outputs for pair of sectors i and j .

- The quantity demanded: the number of units of sector j 's output purchased by sector i .
- The expense: the dollars spent by sector i to buy the quantity demanded of j 's output.

Then, for each sector:

- The price: the price of one unit of the sector's output, in dollars.
- The quantity supplied: the output of the sector, in the sector's units of production.
- The revenue: the output of the sector in dollars.
- The expense: the dollars spent on the ingredients by the sector.

And then, for the local economy as a whole:

- The unemployment rate, as a percentage of the size of the work force.
- The gross domestic product (GDP), in dollars: the total revenue of the economy, excluding the **else** sector, i.e., the “size” of the local economy.
- The consumer price index (CPI), which measures changes in buying power since the start of the simulation.
- The deflated gross domestic product (DGDP), which is simply the GDP divided by the CPI. This gives the current “size” of the local economy in “time 0” dollars.

7.5 Neighborhood Aggregation/Disaggregation

The CGE covers the entire local economy; but Athena generally deals with happenings in particular neighborhoods. Thus, we need to aggregate neighborhood inputs for use by the CGE, and disaggregate CGE outputs down to the neighborhood level.

7.5.1 Neighborhood Aggregation

The relevant figures are the number of workers, the number of consumers, and the production capacity of each neighborhood. The Demographics model outputs the first two, as described in Section 4. The Economic model itself determines the production capacity of each neighborhood as follows:

- The size of the economy is solved at time 0.
- The quantity supplied by the **goods** sector represents the production of the playbox at time 0.
- We allocate that quantity supplied to the neighborhoods by neighborhood labor force size, yielding the production of each neighborhood.
- We assume that each neighborhood is producing at its capacity at time 0.
- We give each neighborhood a “production capacity factor”, or PCF. A PCF of 1.0 represents the neighborhood’s production at time 0. The PCF can be increased or decreased to reflect increases or decreases in the neighborhood’s production capacity.

The total production capacity of the economy is then the sum of the production capacities of the neighborhoods; this limits the size of the economy in the same way that the size of the labor force does.

7.5.2 Neighborhood Disaggregation

It would seem reasonable to disaggregate a number of outputs to the neighborhood level, such as the total production of goods or the wages paid. This rarely turns out to be useful, however, as the only basis for doing so is usually the population size, which was used to do the aggregation to begin with.

Consequently, we currently disaggregate only the Unemployment Rate, resulting in the number of unemployed workers per neighborhood. This turns out to be interesting because the size of the labor force as a fraction of the population varies from group to group and neighborhood to neighborhood, due to the subsistence agriculture percentage (Section 4.3). We then use the number of unemployed workers relative to the neighborhood population to drive the Unemployment situation (Section 4.5).

7.6 Ways to Affect the Economy

In Athena 3, there are four ways to affect the economy:

- Civilian casualties can decrease the number of consumers and workers.
- When displaced from their land, the subsistence population willy-nilly become consumers, and might or might not contribute to the labor force.
- When a civilian group's security decreases, workers can stay at home out of fear, thus reducing the effective size of the labor force.⁷
- Each neighborhood's Production Capacity Factor can be increased or decreased, reflecting the building of new plant or destruction of existing plant, and thereby changing the economy's production capacity.

7.7 Effects of the Economy

There are many ways in which the economy *should* affect Athena; at present, the only implemented effect is that of unemployment on the civilian population (Section 4.5). In addition, the analyst can track the GDP and the CPI, and use them to inform his own interventions.

⁷ This part of the model is known to have problems, which will be resolved in the six-sector model.

8. INFORMATION

Information, its use, and its spread are crucial in the kinds of scenarios Athena has been designed to address. The Information Area includes the following:

- Command and control of troops and organizations
- The spread of information within the civilian population and across the playbox
- Intelligence received by force groups (and hence by actors) from the civilian population
- Information operations: propaganda and other media techniques intended to affect the perceptions of the people in the playbox, so as to:
 - Increase support for an actor
 - Decrease support for an actor
 - Skew the intel available to an actor, thus affecting the activities driven by it.

At present, Athena's information modeling is fairly simple:

- GRAM tracks the cooperation (willingness to give information) of civilian groups with force groups; cooperation has a number of effects across the playbox, and especially on combat.
- GRAM includes a simple model of information flow across the playbox as it applies to attitude effects.

For the rest, information flow is implicit in Athena's models. This is a major area for future work.

Part II: Using Athena

9. INSTALLATION

Athena requires 32-bit Windows, and specifically Windows 7, and is known to operate on 64-bit Windows 7. It is believed to operate on Windows Vista, and may also work on Windows XP.

To install Athena, first acquire the installation wizard from the TRADOC G-2 Intelligence Support Activity (TRISA) at Ft. Leavenworth, Kansas; at time of writing, the contact is Robert Crowson, at Robert.Crowson@us.army.mil. Save the installer to your desktop and execute it in the usual way; the wizard will lead you through the installation process.

9.1 Starting Athena

At the end of the process Athena will be available from the Start menu, and optionally as an icon on your desktop. In addition, you can start Athena with a particular scenario by double-clicking on the scenario file.

9.2 Athena Documentation

The Athena documentation, including this User's Guide, is available on the Windows Start menu. In addition, the Athena application has extensive on-line help; select any help topic from the Help menu.

9.3 Multiple Versions of Athena

Multiple versions of Athena can coexist on one machine, and will have distinct entries on the Windows Start menu. However, there are two caveats.

Double-clicking a scenario file will always invoke the most recently installed version of Athena. This will usually be what you want. If it is not, invoke the correct version, and use its File/Open menu item to open the scenario.

Second, the scenario file's schema frequently changes during Athena development. If you are running development builds (e.g., version 3.0.x or 4.0.x) be aware that a scenario created with one version might not work with a later version, and that you might have to take extra steps to convert it. Release builds (e.g., successive releases of version 3.1.x or 4.1.x) do not have this problem. When installing a new version, always look at the Release Notes (available with the Athena documentation on the Windows Start menu); they will tell you of any version incompatibilities.

10. USING THE ATHENA APPLICATION

This section gives a top-level overview of the Athena application and how to use it.

10.1 Athena Scenario Files

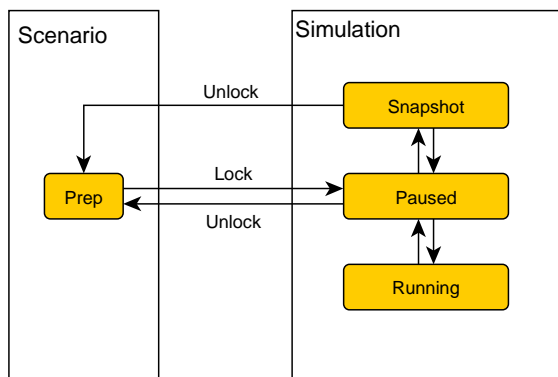
Athena is a document-centric application, like a word processor or spreadsheet application. In Athena's case, the documents are called *scenario files*. Each scenario file includes the data required by a single Athena scenario: neighborhoods, actors, groups, strategies, and other inputs. If saved after the Athena simulation has run forward in time, the scenario file will also include the results.

Athena provides the usual File/New, File/Open, File/Save, and File/Save As menu items; you can also invoke Athena with a specific scenario file by double-clicking the scenario file in the Windows Explorer (but see Section 9.3 if you have multiple versions of Athena installed).

By default, Athena saves scenario files in your Documents\Athena folder. However, you may save them anywhere you like.

10.2 Athena Workflow

Athena operates in several distinct states; what you can do varies from state to state. The states are shown in the following diagram:



When you create a new scenario, Athena is in the *Scenario Preparation* state, called **Prep** for short. In this state you create the scenario, adding neighborhoods, actors, groups, strategies, and so forth. When you've defined the scenario to your liking, you *lock* the scenario, entering the **Paused** state. In this state the simulation is initialized, but time is not advancing. You can examine the initial state of the simulation, and then either return to the **Prep** state or tell Athena to run time forward.

While time is advancing, Athena will be in the **Running** state; in this state, most user input is blocked. After advancing time the appointed amount, Athena will return to the **Paused** state, where you can examine the results. You can then make changes, if you like,⁸ and direct Athena to continue advancing time. Alternatively, you can unlock the scenario, returning the **Prep** state (and losing any simulation results). Once there, you can revise the scenario and start over.

Finally, each time Athena leaves the **Paused** state for the **Running** state it saves a snapshot of the current state of the simulation. Whenever the simulation is **Paused** you can browse through these snapshots, examining the state of the simulation at these earlier points in simulation time. In this **Snapshot** state you cannot make any changes, but you can view all outputs. While viewing a snapshot, you can choose to return to the latest simulation time, or “re-enter the time stream” as of the snapshot time, make changes, and run forward from there. Finally, you can always unlock the simulation and return to the **Prep** state.

Note that if you save the scenario while Athena is **Paused** or in the **Snapshot** state, all simulation results and snapshots will be saved in the scenario file, and will be available when you re-open the file.

10.3 Scenario Mode vs. Simulation Mode

In the **Prep** state you can edit the scenario, but you can't advance simulation time. In the **Paused** state, you can advance simulation time, but you can't edit the scenario. The data and tools required are different in these two states, and Athena's user interface updates to match. In *Scenario Mode*, Athena displays the tools you need to create and edit the scenario. When you lock the scenario, Athena switches to *Simulation Mode*, and displays the tools you need to control simulation time and view the results.

While in Simulation Mode many of the screens used to input scenario data are not available; however, you can still find the scenario inputs using the Detail Browser (Section 11.5). Alternatively you can unlock the scenario, or temporarily switch the GUI back to Scenario Mode using the View menu.

10.4 Viewing Athena Results

Athena provides a great many tools for browsing the Athena results interactively, including a number of bar charts and line plots; these tools will be described in Section 11. The most

⁸ Note that there are many changes (e.g., adding a group) that you can only do in the **Prep** state.

important of these tools is the Detail Browser, which provides a web-browser-like view of all of the scenario data.

It is also possible to access Athena results programmatically. The Athena scenario file is an SQLite database file⁹; SQLite bindings are available for most programming languages. You can browse Athena's SQL database schema by entering the address "my: //rdb" into Athena's **Detail Browser** (Section 11.5).

10.5 The Significant Events Log

As Athena advances simulation time, it creates a log of significant simulation events: tactics executed, deployments, civilian casualties, and so forth. Errors in **EXECUTIVE** tactics and **EXPR** conditions also appear in the significant events log.

To view the significant events log, go to the Detail Browser, and select "Sig. Events" in the Objects tree under the "Overview" heading. In addition, the actor, neighborhood, and group pages all end with a list of the significant events relevant to that entity from the last time advance.

10.6 The Debugging Log

As Athena runs, it creates a low-level log of its activities. This log is primarily intended for use in debugging the application and its models, and so it is not usually visible in the GUI. The information contained in the debugging log may be useful to the analyst at times, however. To view the debugging log, select **View/Scrolling Log** from the Athena menu. This will make the **Log** tab visible; you can then select the **Log** tab to see the actual log. See Section 11.17 for more on the log browser.

10.7 Background Errors

When Athena encounters an unexpected run-time error, it does not usually halt execution. Instead, the error and a complete stack trace are written to the *debugging log*, and the user is notified. This is called a *background error*, or *bgerror* for short.

It is usually wise to save the scenario under a new name after a bgerror, and then restart the application. However, the only consequence of most bgerrors is that a particular feature is unavailable, or unavailable in certain circumstances.

⁹ See sqlite.org for more information.

Background errors should be reported to the development team. The report should include the following items, insofar as this is possible given security considerations:

- The berror and stack trace from the debugging log. This is the text with the orange background.
- The complete text of the log file, if possible. Select **detail** on the **Log** pull-down on the Log Browser toolbar before cutting and pasting the log text.
- A description of what was happening immediately before the berror occurred (if known).
- A detailed procedure for reproducing the error, if known.

It is often wise to contact the development team with the bare fact of the error prior to gathering all of the information, in case it is a known problem.

10.8 Athena Scripting

Athena provides significant support for scripting through the *Athena Executive*, a scripting language processor based on the TCL programming language.¹⁰

10.8.1 Executive Commands

The Athena Executive extends the TCL programming language with a wide variety of *executive commands*; these commands can create and revise neighborhoods, actors, groups, and so forth, lock and unlock the scenario, advance time, and so forth. The complete list of executive commands is given (with full documentation) in the Athena application's on-line help; select "Executive Commands" from the Help menu.

The most important executive command is the `send` command, which is used to send *orders* to Athena. See Section 17 for more about Athena orders.

10.8.2 The Athena Command Line

Executive commands can be entered interactively by the user at the Athena command line. The command line is not displayed by default; to make it available, select "Command Line" from the View menu.

10.8.3 Executive Command Scripts

The user can also place a succession of executive commands in a text file using NotePad or some other text editor; such a file is called a *script*. In addition to Athena's executive

¹⁰ See `www.tcl.tk` for more information.

commands, the file may contain general TCL-language program code, including loops, conditionals, procedures, and so forth. Such a script can be called in a number of ways:

- From the Athena Command Line, using the `call` command.
- When invoking Athena in batch mode (see Section 10.9).
- By **EXECUTIVE** tactics (see Section 15.7).

It is customary to give executive scripts a `".tcl"` extension, e.g., `"myscript.tcl"`.

10.8.4 Scenario Scripts

Although the user interface hides this from the user to a great extent, an Athena scenario is built by sending a sequence of simulation orders to Athena. These orders create and update neighborhoods, groups, and so on. Athena keeps track of the series of orders that leads to its current state; you can save them to disk as a script of `send` commands by using the `export` command at the Athena Command line:

```
> export myscenario.tcl
>
```

This will save the script as `myscenario.tcl` in the same directory as the scenario file. The script can then be executed in the usual way to recreate the existing scenario, or edited by hand to create a slightly different scenario. This provides a convenient way to load an existing scenario into a new version of Athena when using development builds (Section 9.3): export it from the old version, make any necessary changes by hand, and load it into the new version.

Naturally, scenario scripts can also be created by hand or by other software; this can be a convenient way to load data from other databases into Athena. It is simply necessary to use the proper sequence of Athena orders. See Section 17 and the application's on-line help for documentation on the individual orders.

10.9 Batch Mode

Athena can be invoked in *batch mode* from the Windows command line or by other programs. In batch mode Athena has no user interface; instead it initializes itself, runs a script, and then shuts down. Typically the script will advance time and save the scenario under another name. The results can then be acquired by querying the saved scenario file as an SQLite database.

10.9.1 Invoking Athena in Batch Mode

To invoke Athena in batch mode, use the following command line (question marks denote optional arguments):

```
athena.exe -batch -script myscript.tcl ?myscenario.adb?
```

In batch mode:

- No GUI is created.
- A scenario file is loaded, if one is given on the command line.
- A script is executed if the `-script` option is given.
- If results are to be saved, the script must explicitly call the `save` command.
- Athena exits.

If Athena runs with no errors, it will terminate with exit code 0. If any error occurs, details about the error will be written to the file `"error.log"` in the user's working directory, and Athena will terminate with exit code 1.

10.9.2 Simulation Control

In batch mode, the simulation is controlled by a script of executive commands specified using the `-script` option. Documentation for the Athena executive commands is in the Athena help; to view it, invoke Athena interactively, and select Help/Executive Commands from the menu bar.

The following executive commands will be most useful; see the Help for full details, including command syntax:

- `advance`: Advances simulation time.
- `call`: Calls another script.
- `export`: Exports the current scenario as a script.
- `load`: Loads a scenario file.
- `lock`: Locks the scenario so that time can advance.
- `new`: Creates a new, empty scenario.
- `save`: Saves the scenario, including any simulation results.
- `send`: Sends an Athena order.
- `unlock`: Unlocks the scenario, so that basic scenario data can be changed.

10.9.3 Simulation Results

To process simulation results, use the `save` executive command to save a scenario file after advancing time; this file will contain the results, along with all of the other scenario data. The results can be viewed in two ways:

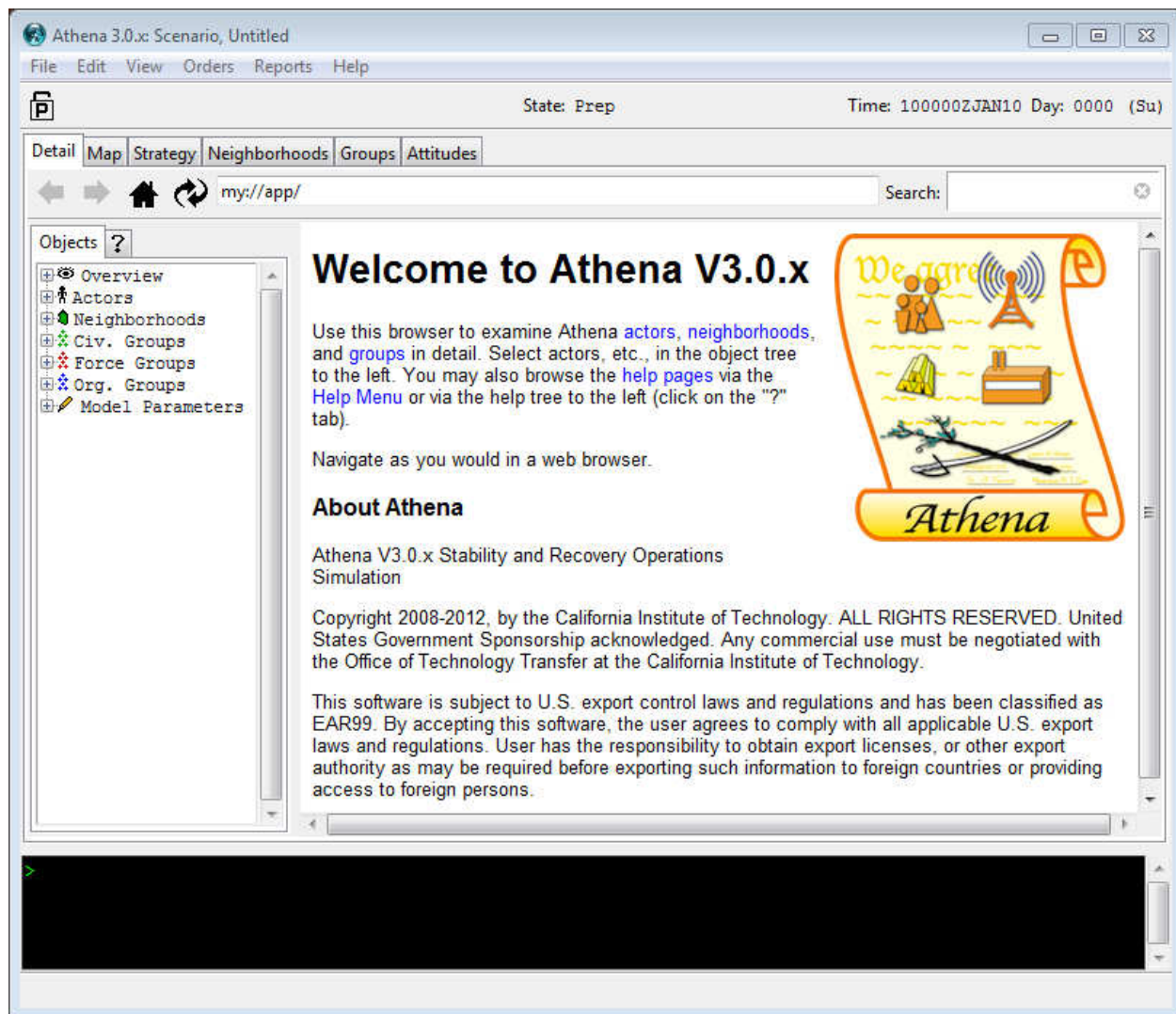
- First, you can open the file in Athena and browse it interactively.

- Second, the scenario file is an SQLite database; any SQLite client can open it and execute queries. You can browse the schema by running Athena interactively and entering "my: // rdb" into the Detail Browser's address bar.

11. THE ATHENA USER INTERFACE

This section describes the parts of the Athena user interface, and how to make use of them.

11.1 The Main Window



The Athena main window consists of the following components:

- A traditional menu bar (Section 11.2)
- The application toolbar, whose contents changes depending on the application mode, Scenario or Simulation (Section 11.3)
- The main body of the window, which contains many tabbed data panes, some with sub-tabs of their own.
- The command line, where executive commands may be entered interactively. The command line is hidden by default. (Sections 10.8 and 11.4)
- The status line, where Athena displays transient messages.

These components are discussed in detail in the following sections.

11.2 The Menu System

The menu items in Athena menu system are documented in detail in the on-line help. This section gives an overview of the top-level menus.

11.2.1 The File Menu

From the **File** menu you can create new scenarios, open existing scenarios, save the scenario, and so forth in the usual way. In addition, you can import maps (Section 11.6), import and export the model parameter database (Section 12.16), and save the Command Line's scrollbar buffer to disk as a text file.

11.2.2 The Edit Menu

The **Edit** menu has the usual Undo/Redo, Cut/Copy/Paste, and Select All menu items.

The Undo/Redo capability operates on the level of individual simulation orders as executed interactively via the GUI. For example, if you use the Update Civilian Group order to change a civilian group's base population and demeanor, you can undo the entire change. Athena maintains a deep undo stack; the stack is cleared when the scenario is saved, when the scenario is locked, and when simulation time starts to advance.

The Cut/Copy/Paste operations apply only to text, and apply to any field in which text is edited. In addition, it is possible to select and copy text from the **Detail Browser**. It is not possible to cut, copy, and paste other items of data, e.g., tactics and conditions.

11.2.3 The View Menu

The **View** menu allows the user to determine which application components are visible at any given time.

Scenario Mode vs. Simulation Mode. The primary distinction is between Scenario Mode, in which the controls and tabs required for scenario editing are visible, and Simulation Mode, in which the controls and tabs required when running the simulation are visible. The application switches automatically between these two modes when the scenario is locked and unlocked; the user can switch between them temporarily using the **View/Scenario** and **View/Simulation** menu items.

View a Specific Tab. The second section of the **View** menu lists all of the currently visible tabs and sub-tabs. You can have Athena go directly to a particular tab by selecting the corresponding menu item.

Optional Components. By default the Athena command line and the **Firings, Orders, and Log** tabs are not displayed. The user can make them visible by selecting the relevant item from the bottom of the **View** menu.

11.2.4 The Orders Menu

The **Orders** menu contains submenus and menu items for almost all of the simulation orders supported by Athena. Selecting the menu item pops up the order dialog for the given order. If an order cannot be used in the current simulation state, the related menu item will be disabled. The orders themselves are documented in the on-line help; select **Help/Orders** from the menu.

Some orders are only available from other parts of the GUI; for example, it's necessary to attach a condition to a tactic from within the **Strategy Browser**. Consequently, these orders do not appear on the **Orders** menu.

11.2.5 The Reports Menu

A report is simply a page in the **Detail Browser** that displays data about the simulation; selecting a report from the **Reports** menu pops up a dialog that allows the user to filter and otherwise customize the data included in the report. The specific reports are documented in the on-line help; select **Help/Orders** from the menu, and then click on **Report Orders** in the **Detail Browser**.

The most useful report is the **Contribution to Satisfaction** report, which lists for any satisfaction curve and time period the drivers which made the largest contributions to the curve, and the size of each driver's contribution.

11.2.6 The Help Menu

The **Help** menu is a convenient way to access Athena's on-line help, which is displayed in the **Detail Browser**. You can also see an outline of all of the help topics by clicking on the "?" tab in the **Detail Browser**'s sidebar.


You can browse the on-line help and use the application at the same time by selecting **File/New Detail Browser...** from the menu. This will pop up a new window containing only a **Detail Browser**; then, select the help topic from the sidebar.

11.3 The Main Toolbar

The content of the main toolbar changes depending on whether the application is in Scenario Mode or Simulation Mode.






11.3.1 The Scenario Mode Toolbar

In Scenario Mode, the main toolbar contains the following controls:

| Control | Description |
|---|---|
|  | Lock Scenario Preparation. When pressed, this button locks the scenario. All sanity checks are done, and the application enters Simulation Mode. |
| State: Prep | The toolbar always shows the current simulation state. |
| Time: 100000ZJAN10 Day: 0000 (Su) | The toolbar always shows the current simulation time. In Scenario Mode, this is time 0, the simulation time we will have when the simulation is locked. The mapping from the simulation time in integer days to zulu-time strings can be changed by selecting Orders/Simulation/Set Start Date... from the application menu. |

11.3.2 The Simulation Mode Toolbar

In Simulation Mode, the main toolbar contains the following controls:

| Control | Description |
|---|--|
|  | Unlock Scenario Preparation. When pressed, this button unlocks the scenario, returning to Scenario Preparation. Any simulation results are lost. |
|  | Run Simulation. When pressed, the simulation begins to run. Time for the interval indicated in the Duration of Run pull-down. |
|  | Pause Simulation. While time is advancing, the Run Simulation button turns into a pause button. When pressed, the simulation will pause. |
|  | Leave Snapshot Mode. While in snapshot mode, the Run Simulation button turns into a rewind button. When pressed, the simulation will pause at the time of the currently selected snapshot. Changes can then be made and time can be advanced; however, any simulation results later than the selected time will be lost. |
| 7 days | Duration of Run. Determines the length of the next time-step. |
|  | VCR Buttons. These buttons are used to step through the available snapshots. The rightmost button returns the simulation to the latest simulation time, leaving snapshot mode. |
| State: Paused | The toolbar always shows the current simulation state. |
| Time: 240000ZJAN10 Day: 0014 (Su) | The toolbar always shows the current simulation time. |

11.4 The Command Line

The Athena Command Line allows the user to enter executive commands interactively; see Section 10.8 for more information about executive commands. For example, the `=` command can be used as a calculator:

A screenshot of a terminal window with a black background and green text. The text shows a prompt character followed by the command `= 1 + 2`, and on the next line, the output `3`. A vertical scrollbar is visible on the right side of the terminal window.

```
> = 1 + 2
3
>
```

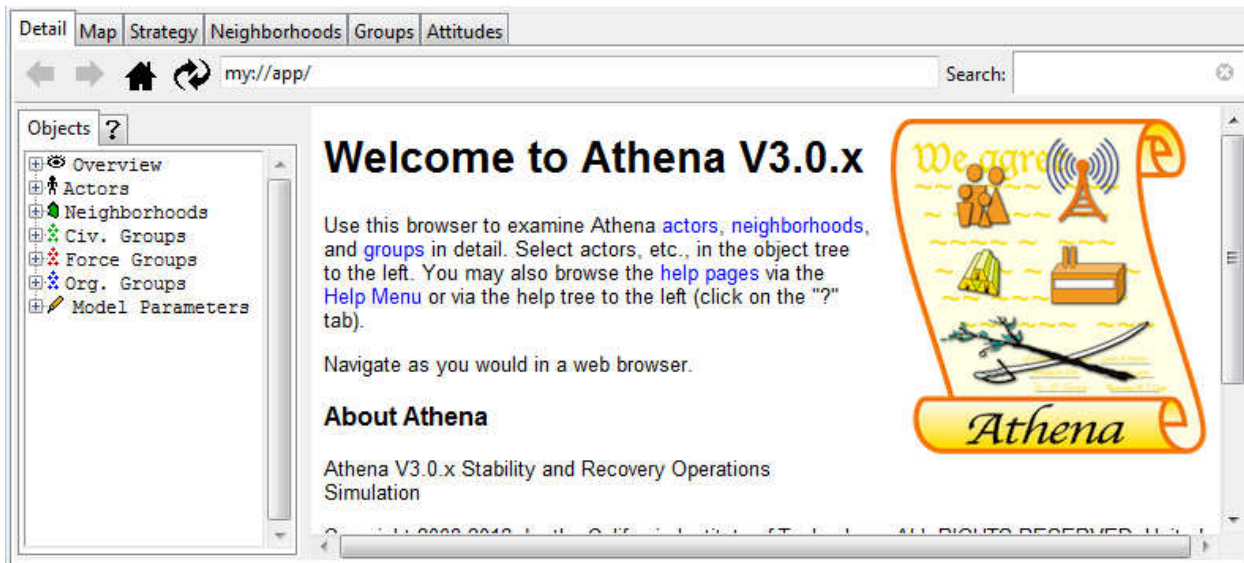
The Command Line is not visible initially; to make it visible, select **View/Command Line** from the menu.

While the cursor is in the Command Line, the up and down cursor keys can be used to step through and select previously entered commands.

By default, the Command Line retains 500 lines of output; this output can be browsed using the scrollbar, and can be saved to disk by selecting **File/Save CLI Scrollback Buffer...** from the menu.

11.5 The Detail Browser

The **Detail Browser**, contained on the **Detail** tab in the main window, is the central part of the Athena user interface. It is the primary spot for viewing the scenario and the results.



In form, the **Detail Browser** looks much like a web browser. It displays pages of HTML-formatted data, each of which has its own URL. The user can navigate the data by clicking on links and by use of the back and forward buttons, and by entering specific URLs into the browser's address bar. However, the **Detail Browser** does not implement true web access: it will only display data from within the application, not data from external web pages.

The **Browser** has several distinctive parts:

- The Sidebar, which provides quick links to many of the available pages.
- The toolbar, which provides the usual back, forward, home, and reload buttons, the address bar, and a search box.
- The data pane, in which data pages are displayed.

By selecting **File/New Detail Browser...** from the menu, the user can pop up a window containing only a **Detail Browser**; in this way it is possible to look at two pages at the same time, or to look at a page while using some other portion of the user interface.

11.5.1 Detail Browser Sidebar

The Sidebar has two tabs: the Objects tab and the "?" or Help tab. The Objects tab contains a tree of simulation objects and other topics for which pages exist. Selecting an entry in the tree

displays the related page in the data pane. Note that the data available for an entry often depends on the state of the simulation.

The “?” tab includes a tree of all of the topics in the on-line help. Selecting a topic displays the help page in the browser.

11.5.2 Detail Browser Search Box

The Search Box provides an easy way to search the on-line help for information about a particular topic. Unfortunately, it does not search any of the rest of pages available in the browser. To search for a topic, enter a string in the search box. The usual operators are available; for example, enclose a phrase in double-quotes to get an exact match.

11.5.3 Context Menu

Right-clicking on the data pane pops up the **Detail Browser**'s context menu, which contains the following items:

Save to Disk...

Saves the currently displayed page to disk as an HTML file.

View in System Web Browser...

Saves the currently displayed page to disk as an HTML file, and displays it in the system's default web browser. This can be a convenient way to print a browser page.

View Source...

Displays the HTML source in a pop-up window.

11.5.4 Useful URLs

Most of the pages of interest are included in the Sidebar. However, there are some URLs which are not, or which are otherwise useful to know.

my://app

This is the application's home page.

my://help

This is the root of the on-line help tree.

my://rdb

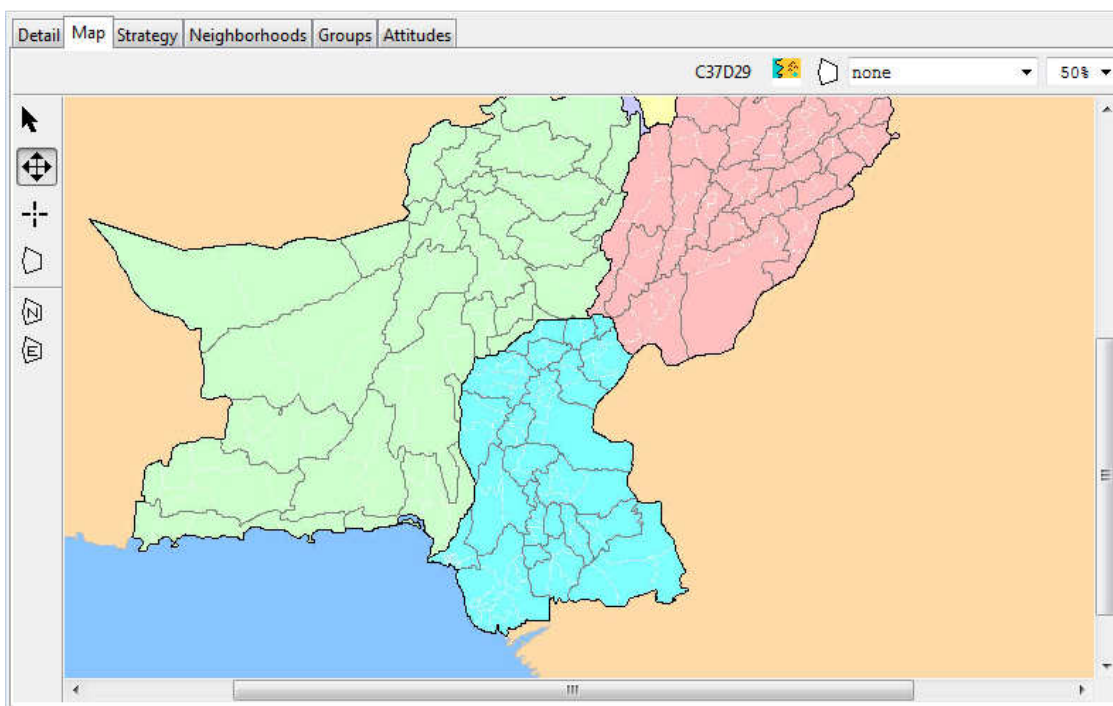
This URL is the root of a tree of pages concerning the schema and content of Athena's run-time database (RDB). If you wish to query the RDB or a scenario file directly using SQLite, this is a good place to start exploring.

Adding `/urlhelp` to any of these root URLs brings up a page that describes all of the URLs that can be built from that root. For example, browsing `my://rdb/urlhelp` will tell you that you can type `my://rdb/schema/civgroups` into the address bar to browse the SQL schema of the `"civgroups"` table.

The URL of the current page is always displayed in the **Detail Browser's** address bar; and when the mouse pointer is over a link, the linked URL will appear momentarily on the window's status line.

11.6 The Map Browser

The Map Browser, contained on the **Map** tab, is a tool for visualizing the playbox. It displays neighborhoods, units, and environmental situations; and the neighborhood polygons can be colored to display various neighborhood display variables.



11.6.1 The Map Proper

The main part of the Map Browser displays the actual map. By default, this is simply a blank canvas on which neighborhoods can be drawn. Alternatively, any desired map image can be imported as a map background by selecting **File/Import Map...** from the menu.

11.6.2 Map Reference Strings

Because Athena is designed to use any map image the analyst may have on hand, from a high-quality scan of a printed map to a snapshot of a map hand-drawn on a piece of paper, it cannot assume that maps are properly geo-referenced, and so cannot use geographic coordinates or grid reference strings.


Instead, Athena uses (X,Y) coordinates in the range 0 to 999 for both axes. The origin is at the upper left; positive X runs from left to right and positive Y from top to bottom. These coordinates are then mapped to the pixel coordinates of the map, with the map's longer dimension mapping to the range 0 to 999 and its shorter dimension mapping to a

proportionally smaller range. For example, the lower right corner of a map 1000 pixels wide by 500 pixels high would have numeric map coordinates (999,499).


These numeric coordinates are used internally; for input and output they are converted into *map reference strings*. A map reference string is a six-character string that represents a point on the *scenario's* map. It has the form **AnnAnn** where *A* is a letter from "A" to "K" (excluding "I") and *n* is a digit, "0" to "9". Thus, the location (123,456) has reference string "B23E56".

11.6.3 Neighborhood Polygons

As part of creating a neighborhood, the user must supply a neighborhood polygon. The steps are as follows:

1. Go to the **Map** tab
2. Click on the **Create Neighborhood** tool, .
3. Enter the neighborhood information into the fields in the Create Neighborhood dialog.
4. Select the Reference Point field, then click on the neighborhood's reference point on the map. The map reference string will appear in the field.
5. Select the Polygon field, and then draw the polygon on the map:
 - a. The first time you click on the map you will start to draw a line.
 - b. Click on the vertices of the desired polygon, proceeding around the edge of the neighborhood.
 - c. The mouse pointer will automatically snap to the vertices of existing neighborhoods. This makes easy to create neighborhoods with shared borders.
 - d. At the end, either click on the first point again, or simply double-click on the last point.
 - e. The polygon coordinates will appear in the field.
6. To change a reference point or polygon, simply repeat the steps given above.

11.6.4 Unit and Environmental Situation Icons

The Map Browser displays unit and environmental situation icons. To see more information about an icon, select the Browse Tool, , and click on the icon. Information will appear on the status line.

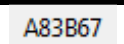



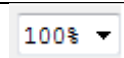
Environment situation icons indicate the state of the situation. If the situation has not yet had its attitude effects assessed (i.e., it is brand new, and time has not advanced), it will display with red text on a white background. If it has been assessed and is not resolved, it will display with red text on a yellow background. If it has been resolved, it will display with green text on a white background.

To move an icon, Control-Click on it and drag it to its new location. Unit icons can only be moved within their neighborhood. Environmental situation icons can be moved between

neighborhoods provided that they have not yet been assessed; after that, they must remain in the same neighborhood.




11.6.5 Map Browser: Main Toolbar




The main toolbar controls determine how the map data is displayed:

| Control | Description |
|---|---|
|  | The map reference string under the mouse pointer. |
|  | Extended Scroll Region. Most map images are rectangular; Athena's map coordinates describe a square region. By default the Map Browser displays only the map proper; if this button is latched, the Map Browser will scroll to the addressed region that is off the bottom or right side of the map image. This can be a useful place to draw remote neighborhoods, i.e., staging areas or communities in other parts of the world. |
|   | Opaque Fill. By default, neighborhood polygons are transparent. If this button is latched, they are filled with some color. The color can simply be white, or it can vary across the neighborhoods to visualize some neighborhood display variable. The variable to use is selected from the pull-down. As shown, the neighborhoods will be colored according to the neighborhood mood, with positive satisfactions ranging from white to bright green and negative satisfactions ranging from white to bright red. |
|  | Zoom factor. Zoom in and out on the map by selecting the desired zoom factor. |

11.6.6 The Secondary Toolbar

The secondary map toolbar runs down the left side of the map, and contains controls for interacting with the map and its contents. The controls are as follows:

| Control | Description |
|---|---|
|  | Browse Tool. Use this to click on items on the map. |
|  | Pan Tool. With this tool selected, the user can scroll the map by clicking and dragging. |
|  | Point Tool. Use this tool to select map coordinates for entry into order dialogs. <i>Note: It is rarely necessary to select this tool explicitly, as Athena selects it automatically when the user edits a field requiring a map coordinate to be entered.</i> |

| Control | Description |
|---|--|
|  | Draw Polygon Tool. Use this tool to draw neighborhood polygons and enter their coordinates into order dialogs. <i>Note: It is rarely necessary to select this tool explicitly, as Athena selects it automatically when the user edits a field requiring a map coordinate to be entered.</i> See Section 11.6.3 for information on how to draw neighborhood polygons. |
|  | Create Neighborhood. Pops up the Create Neighborhood dialog. |
|  | Create Environmental Situation. Pops up the Create Environmental Situation dialog. |

11.6.7 Context Menus

Neighborhood polygons, unit icons, and environment situation icons all of context menus; access them by right-clicking the mouse on the polygon or icon.

The neighborhood context menu has the following options:

Create Environmental Situation

Pops up the Create Environmental Situation dialog with the right-clicked location filled in.

Attrit Civilians in Neighborhood

Pops up the Magic Attrit Neighborhood dialog with the neighborhood name filled in. This allows the user to cause casualties to all groups in the neighborhood.

Bring Neighborhood to Front

Brings the polygon to the top of the neighborhood stacking order.

Send Neighborhood to Back

Sends the polygon to the back of the neighborhood stacking order.

The unit icon context menu has the following options:

Magic Attrit Unit

Pops up the Magic Attrit Unit dialog with the unit name filled in. This allows the user to cause casualties to the specific group.

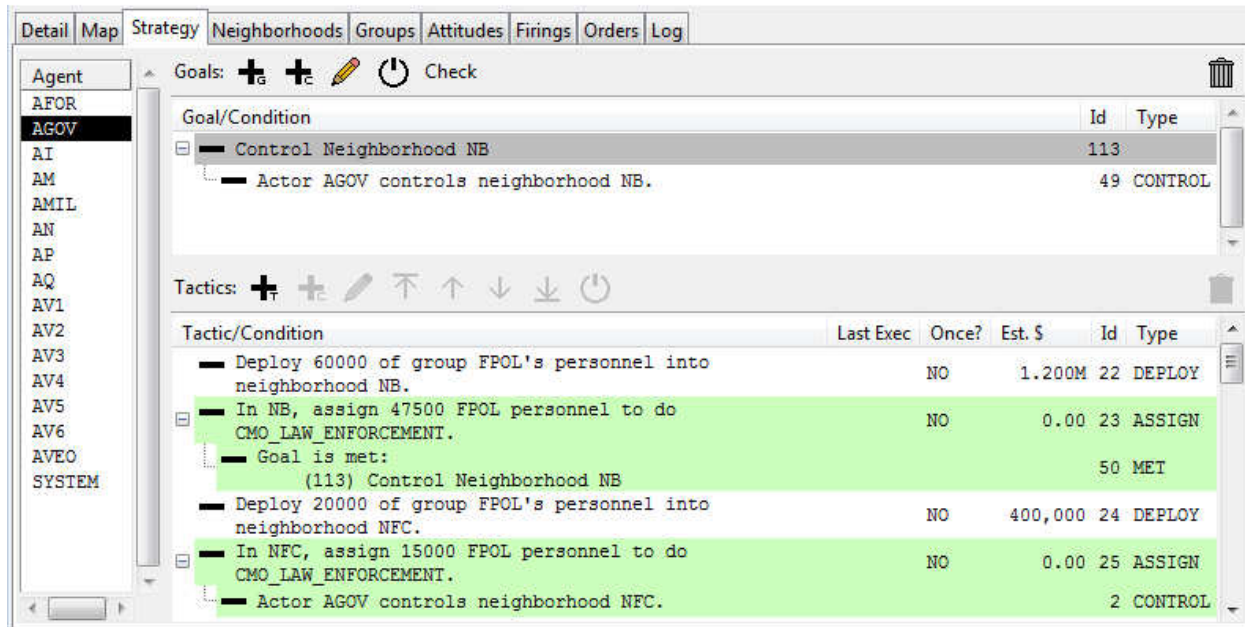
The environmental situation context menu has the following options:

Update Situation

Pops up the Update Environmental Situation order. This allows the user to modify an environmental situation, provided that it has not yet been assessed.

11.7 The Strategy Browser

The Strategy Browser is where actor strategies are created, edited, and viewed.



The browser has three major sections:

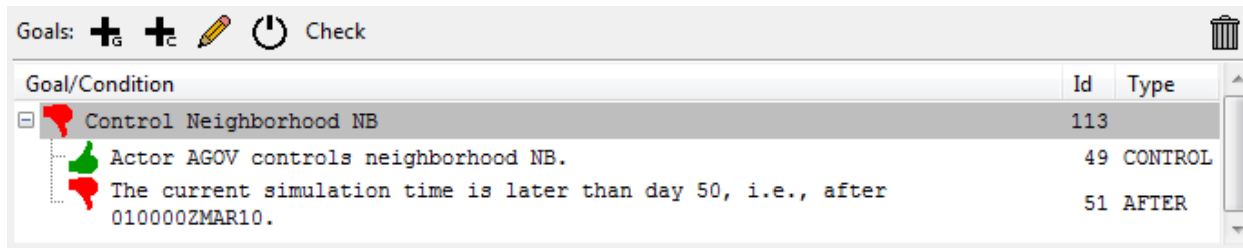
- The Agent list
- The Goal/Condition pane
- The Tactic/Condition pane

11.7.1 The Agent List

The Agent list includes the IDs of all defined actors plus the special **SYSTEM** agent. Select an agent from the list to see its strategy.

11.7.2 The Goal/Condition Pane

The Goal/Condition Pane displays the agent's goals. The conditions which define each goal are listed underneath; click the open/close box to the left of each goal to show or hide the attached conditions.



The pane displays, for each goal, a status icon, the goal's narrative text, and the goal ID, which is used for internal bookkeeping. In addition, it displays the same things for each condition, along with the condition's type name. The status icons are as follows:

| Icon | Description |
|------|--|
| | It is not known whether the goal or condition is met or unmet, i.e., it has not been assessed, it's disabled, etc. |
| | The goal or condition was not met when the strategy was last assessed. |
| | The goal or condition was met when the strategy was last assessed. |

To create and edit goals and their attached conditions, use the controls on the Goal/Condition pane's toolbar:

| Control | Description |
|---------|---|
| | Add Goal. Pops up the Create Goal dialog. |
| | Add Condition. Pops up the Create Condition dialog, allowing the user to attach a new condition to the currently selected goal. |
| | Edit. Pops up the appropriate dialog for editing the currently selected goal or condition. |
| | Toggle Goal State. Enables or disables the currently selected goal. When a goal is disabled, it's as though it doesn't exist. |
| | Sanity Check. Forces a sanity check of all strategies. Tactics and conditions are guaranteed to be "sane" when they are created; but if groups or other entities are deleted after a tactic or condition that references them has been created, the tactic or condition is now in error. This is also checked automatically when the scenario is locked. |
| | Delete. Deletes the currently selected goal or condition. |

11.7.3 The Tactic/Condition Pane

The Tactic/Condition pane shows the agent's tactics in priority order, highest first. The conditions attached to each tactic are listed beneath it; click the open/close box to the left of each tactic to show or hide the attached conditions.

| Tactic/Condition | Last Exec | Once? | Est. \$ | Id | Type |
|---|---------------|-------|---------|----|--------|
| ✓ Deploy 60000 of group FPOL's personnel into neighborhood NB. | 1000000ZJAN10 | NO | 1.200M | 22 | DEPLOY |
| ▬ In NB, assign 47500 FPOL personnel to do CMO_LAW_ENFORCEMENT. | | NO | 0.00 | 23 | ASSIGN |
| ! Goal is met: (113) Control Neighborhood NB | | | | 50 | MET |

The pane displays, for each tactic, a status icon, a narrative description of the tactic, the zulu-time at which it was last executed, whether it is intended to be executed only once, the estimated cost (if known), the tactic's ID number, and the tactic's type. For each condition it displays a status icon, a narrative description of the condition, the condition's ID number, and the condition type.

The status icons are as follows:

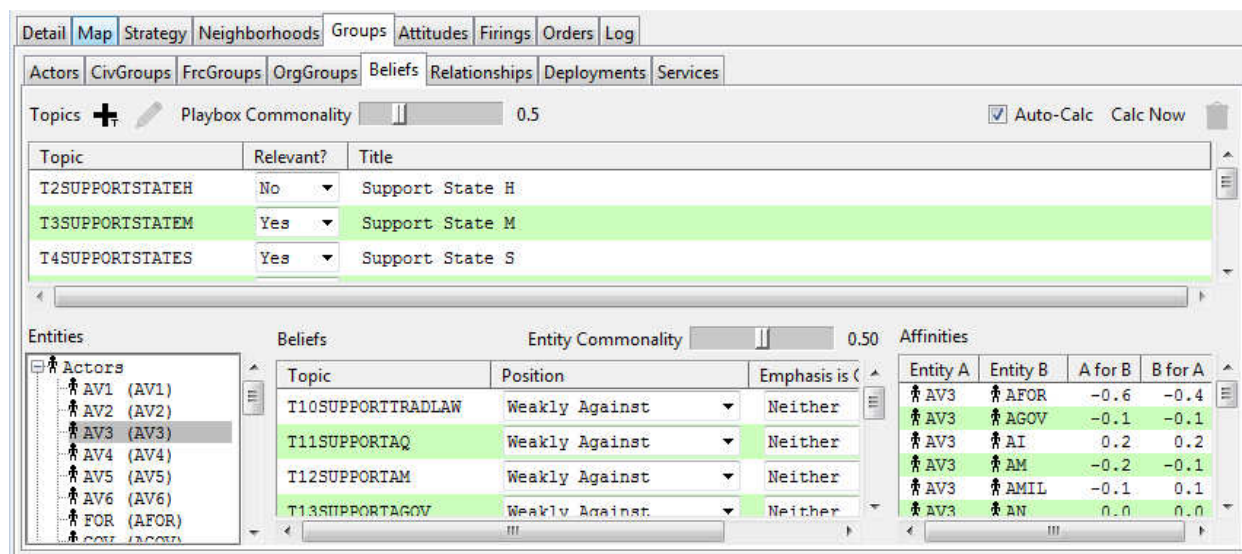
| Icon | Description |
|------|--|
| ▬ | For tactics, the tactic was not executed at the last strategy execution. For conditions, it is not known whether the condition is met or unmet, i.e., it has not been assessed, it is disabled, etc. |
| ✓ | For tactics only, the tactic was executed at the last strategy execution. |
| ! | For conditions only, the condition was not met when the strategy was last assessed. |
| ✓ | For conditions only, the condition was met when the strategy was last assessed. |

To create and edit tactics and their attached conditions, use the controls on the Tactic/Condition pane's toolbar:

| Control | Description |
|---------|--|
| + | Add Tactic. Pops up the Create Tactic dialog. |
| + | Add Condition. Pops up the Create Condition dialog, allowing the user to attach a new condition to the currently selected tactic. |
| | Edit. Pops up the appropriate dialog for editing the currently selected tactic or condition. |
| ↑ ↓ ↕ | Set Priority. These buttons are used to adjust a tactic's priority relative to the agent's other tactics. They move the tactic to the top, up one slot, down one slot, and to the bottom, respectively. |
| ⏻ | Toggle Tactic State. Enables or disables the currently selected tactic. When a tactic is disabled, it's as though it doesn't exist. |
| 🗑 | Delete. Deletes the currently selected tactic or condition. |

11.8 The Belief System Browser

The Belief System browser is found on the **Groups/Beliefs** tab. It is used to create and edit the belief systems of the actors and civilian groups in the playbox (Section 5.1). This browser differs from most of the others in that you can change input values directly in the browser, without popping up a dialog box.



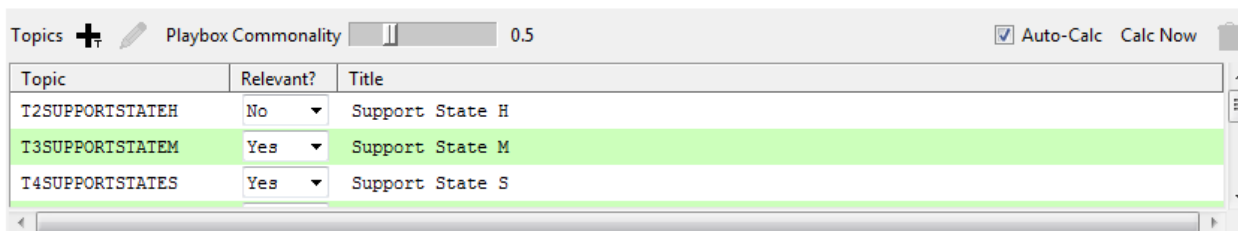
The browser has four major areas:

- The Topics Pane
- The Entities Tree
- The Beliefs Pane
- The Affinities Pane

Note that you can click and drag the border between any pair of these areas to make an area larger or smaller.

11.8.1 The Topics Pane

The Topics pane is where the user creates, edits, and views the set of topics about which actors and civilian groups have beliefs.



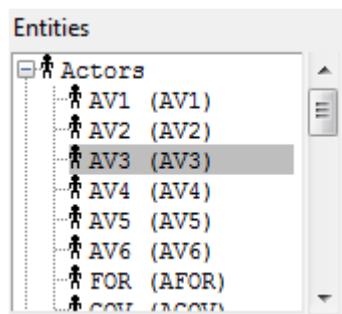
For each topic, the pane shows the topic ID, whether or not the topic is relevant, and the topic's title. The relevance flag is simply a convenient way of enabling or disabling a topic while assembling belief systems, so as to see its effect. To edit a topic's title, click on the title.

The Topic pane has the following controls on its toolbar:

| Control | Description |
|---|--|
| | Add Topic. Pops up the Create Topic dialog. |
| | Edit Topic. Pops up the Update Topic dialog, allowing the user to edit this topic. |
| Playbox Commonality 0.5 | Playbox Commonality. Allows the user to adjust the playbox commonality. |
| <input checked="" type="checkbox"/> Auto-Calc | Auto-Calc. If the box is checked, Athena will automatically compute all affinities any time any part of a belief system is changed. For large scenarios, this quickly becomes annoying. |
| Calc Now | Calculate Now. Forces a recalculation of affinities. Use this after a collection of changes when Auto-Calc is unchecked. |
| | Delete Topic. Deletes the currently selected topic. |

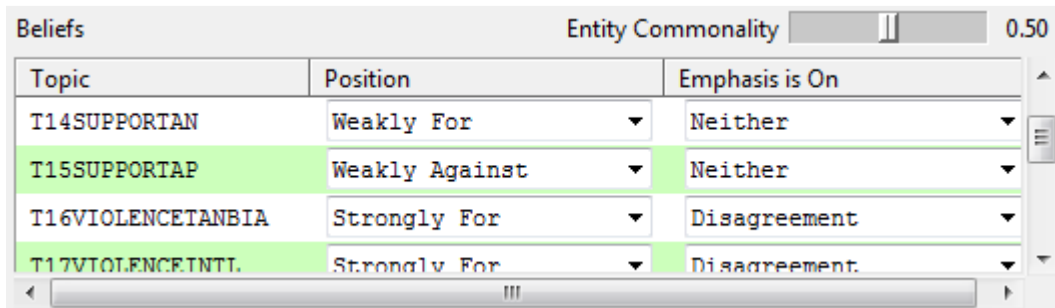
11.8.2 The Entities Tree

The Entities tree displays a tree of the actors and civilian groups in the scenario. Select an actor or group to see its belief system in the Beliefs pane and its affinities in the Affinities pane.



11.8.3 The Beliefs Pane

The Beliefs pane is where the user actually sets a particular actor or group's beliefs.

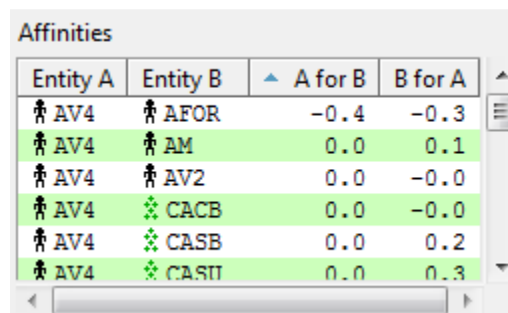


| Beliefs | | | Entity Commonality <input type="text" value="0.50"/> |
|-------------------|----------------|----------------|--|
| Topic | Position | Emphasis is On | |
| T14SUPPORTAN | Weakly For | Neither | |
| T15SUPPORTAP | Weakly Against | Neither | |
| T16VIOLENCETANBIA | Strongly For | Disagreement | |
| T17VIOLENCEINTL | Strongly For | Disagreement | |

It lists the topics in the same order as the Topics pane, and allows the user to select the entity's position and emphasis on each by clicking on the pull-downs. The entity's commonality with the playbox can be adjusted using the Entity Commonality slider on the pane's toolbar.

11.8.4 The Affinities Pane

The Affinities pane shows the affinities of the entity selected in the Entities tree, both its affinities for the other entities, and their affinities for it. As shown, you can sort on any column by clicking on the column header.

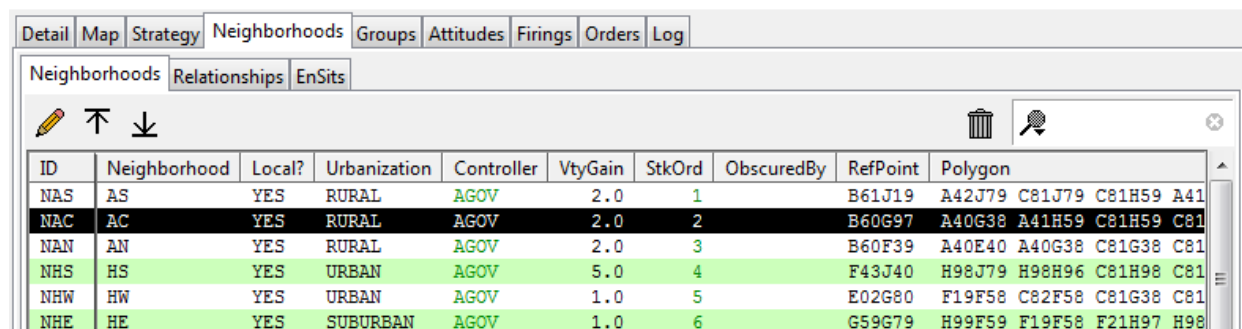


| Affinities | | | |
|------------|----------|---------|---------|
| Entity A | Entity B | A for B | B for A |
| AV4 | AFOR | -0.4 | -0.3 |
| AV4 | AM | 0.0 | 0.1 |
| AV4 | AV2 | 0.0 | -0.0 |
| AV4 | CACB | 0.0 | -0.0 |
| AV4 | CASB | 0.0 | 0.2 |
| AV4 | CASU | 0.0 | 0.3 |

11.9 The Neighborhoods Tab

The **Neighborhoods** tab has a number of sub-tabs which display information about the neighborhoods in the playbox. The precise set of sub-tabs changes when Athena goes from Scenario mode to Simulation mode.

These sub-tabs all have the same form: a tabular display of data with a toolbar. For example, the **Neighborhoods/Neighborhoods** tab shows the neighborhood definitions:



| ID | Neighborhood | Local? | Urbanization | Controller | VtyGain | StkOrd | ObscuredBy | RefPoint | Polygon |
|-----|--------------|--------|--------------|------------|---------|--------|------------|----------|--------------------------|
| NAS | AS | YES | RURAL | AGOV | 2.0 | 1 | | B61J19 | A42J79 C81J79 C81H59 A41 |
| NAC | AC | YES | RURAL | AGOV | 2.0 | 2 | | B60G97 | A40G38 A41H59 C81H59 C81 |
| NAN | AN | YES | RURAL | AGOV | 2.0 | 3 | | B60F39 | A40E40 A40G38 C81G38 C81 |
| NHS | HS | YES | URBAN | AGOV | 5.0 | 4 | | F43J40 | H98J79 H98H96 C81H98 C81 |
| NHW | HW | YES | URBAN | AGOV | 1.0 | 5 | | E02G80 | F19F58 C82F58 C81G38 C81 |
| NHE | HE | YES | SUBURBAN | AGOV | 1.0 | 6 | | G59G79 | H99F59 F19F58 F21H97 H98 |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns and controls listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Neighborhoods/Neighborhoods

Scenario Mode Only. Displays the definition of each neighborhood, with controls to edit, delete, and changing the stacking order of the neighborhoods.

Neighborhoods/Relationships

Scenario Mode Only. Displays the proximity and effects delay between each pair of neighborhoods, with controls to edit these values.

Neighborhoods/Security

Simulation Mode Only. Displays the security of each group in each neighborhood, along with the components of security in the security computation.

Neighborhoods/Activity

Simulation Mode Only. Displays a list of all group activities being done in the neighborhoods of the playbox, including the coverage fractions, the effective personnel, and the resulting activity situation types.

Neighborhoods/ActSits

Simulation Mode Only. Displays a list of all activity situations resulting from the group activities listed on the previous tab, with their attributes.

Neighborhood/DemSits

Simulation Mode Only. Displays a list of all demographic situations currently existing in the playbox, with their attributes.

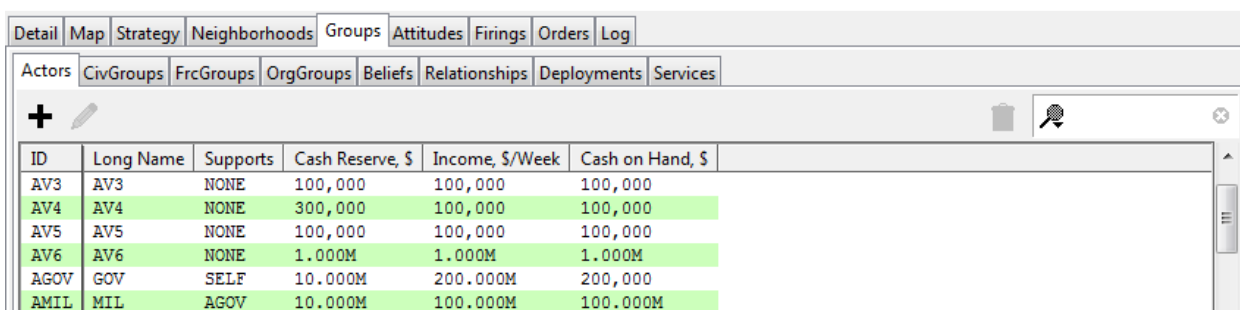
Neighborhood/EnSits

Displays the existing environmental situations, with their attributes. There are controls to create, edit, resolve, and delete situations.

11.10 The Groups Tab

The **Groups** tab has a number of sub-tabs which display information about the actors and groups in the playbox. All of the **Groups** tab's data is for Scenario mode, so the tab does not appear at all in Simulation mode.

All of the sub-tabs have the same form: a tabular display of data (with the exception of the Belief Systems Browser, on the **Groups/Beliefs** tab, which is described in Section 11.8). For example, the **Groups/Actors** tab shows the actor definitions:



The screenshot shows the Athena interface with the 'Groups' tab selected. Under the 'Groups' tab, the 'Actors' sub-tab is active. The table displays the following data:

| ID | Long Name | Supports | Cash Reserve, \$ | Income, \$/Week | Cash on Hand, \$ |
|------|-----------|----------|------------------|-----------------|------------------|
| AV3 | AV3 | NONE | 100,000 | 100,000 | 100,000 |
| AV4 | AV4 | NONE | 300,000 | 100,000 | 100,000 |
| AV5 | AV5 | NONE | 100,000 | 100,000 | 100,000 |
| AV6 | AV6 | NONE | 1.000M | 1.000M | 1.000M |
| AGOV | GOV | SELF | 10.000M | 200.000M | 200,000 |
| AMIL | MIL | AGOV | 10.000M | 100.000M | 100.000M |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns and controls listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Groups/Actors

Displays the actors defined in the playbox, with controls to create, edit, and delete them.

Groups/CivGroups

Displays the civilian groups defined in the playbox, with controls to create, edit, and delete them.

Groups/FrcGroups

Displays the force groups defined in the playbox, with controls to create, edit, and delete them.

Groups/OrgGroups

Displays the organization groups defined in the playbox, with controls to create, edit, and delete them.

Groups/Beliefs

Displays the Belief System Browser, where belief systems are created and edited. See Section 11.8.

Groups/Relationships

Displays the group-to-group relationships. By default, these are simply the affinities computed from the group and actor belief systems; however, the user can override any relationship using the edit control on the toolbar.

Groups/Deployments

Displays the status quo deployments, with controls for editing them. See Section 12.12.

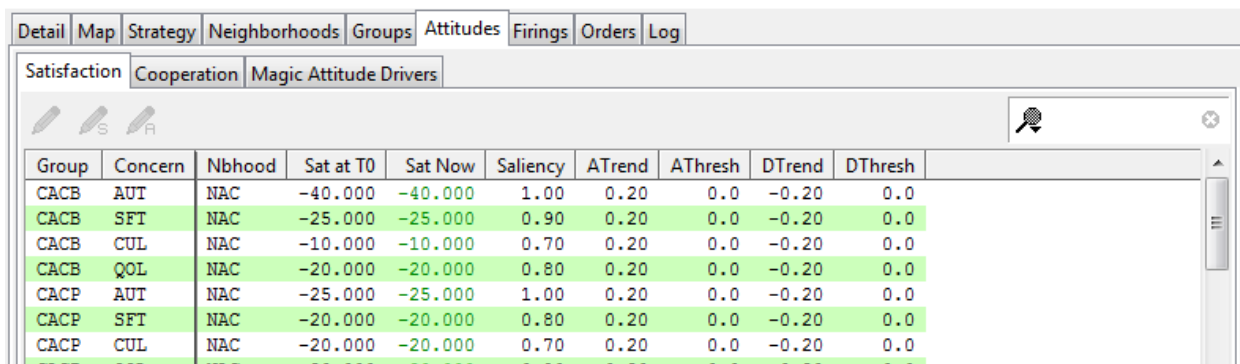
Groups/Services

Displays the status quo funding for ENI services, with controls for editing the funding levels. See Section 12.13.

11.11 The Attitudes Tab

The **Attitudes** tab has a number of sub-tabs which display information about the satisfaction and cooperation of the groups in the playbox.

All of the sub-tabs have the same form: a tabular display of data. For example, the **Attitudes/Satisfaction** tab shows the initial satisfaction curve definitions:



| Group | Concern | Nbhod | Sat at T0 | Sat Now | Saliency | ATrend | AThresh | DTrend | DThresh |
|-------|---------|-------|-----------|---------|----------|--------|---------|--------|---------|
| CACB | AUT | NAC | -40.000 | -40.000 | 1.00 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACB | SFT | NAC | -25.000 | -25.000 | 0.90 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACB | CUL | NAC | -10.000 | -10.000 | 0.70 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACB | QOL | NAC | -20.000 | -20.000 | 0.80 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACP | AUT | NAC | -25.000 | -25.000 | 1.00 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACP | SFT | NAC | -20.000 | -20.000 | 0.80 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACP | CUL | NAC | -20.000 | -20.000 | 0.70 | 0.20 | 0.0 | -0.20 | 0.0 |
| CACP | QOL | NAC | -20.000 | -20.000 | 0.80 | 0.20 | 0.0 | -0.20 | 0.0 |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns and controls listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Attitudes/Satisfaction

Displays the initial satisfaction levels, saliency, and trends for all groups and concerns, with controls to adjust or edit them. In Simulation Mode, it also displays the current satisfaction levels as they change over time.

Attitudes/Cooperation

Displays the initial cooperation levels and trends for all civilian/force group pairs, with controls to adjust or edit them. In Simulation Mode, it also displays the current cooperation levels as they change over time.

Attitudes/NbhodCoop

Simulation Mode Only. Displays the cooperation of each neighborhood with each force group, at time 0 and at the current time. This is simply the average cooperation across the groups in the neighborhood.

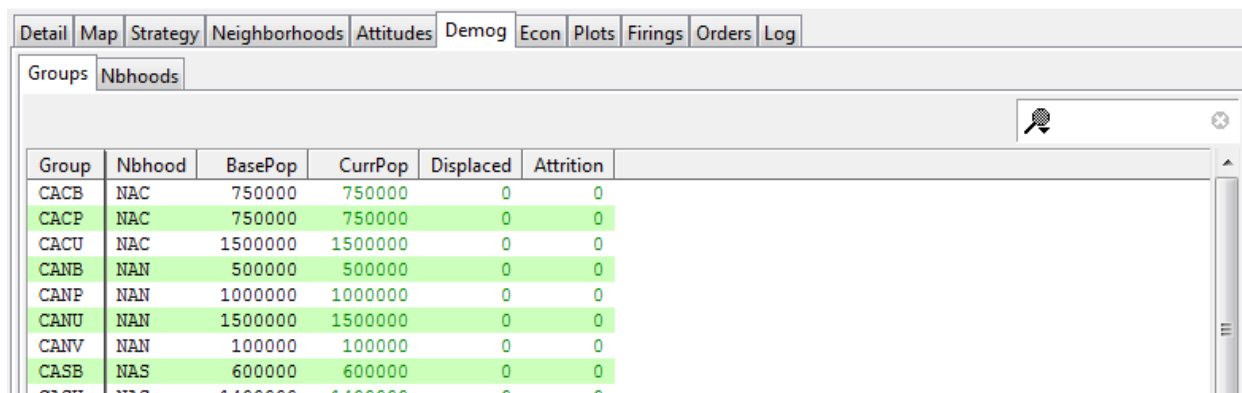
Attitudes/Magic Attitude Drivers

Displays the attributes of all currently defined magic attitude drivers, and allows drivers to be created and edited. See Section 5.8.

11.12 The Demog Tab

The **Demog** tab has two sub-tabs which display the demographics of the playbox by group and by neighborhood. This tab is shown only in Simulation Mode.

Both of the sub-tabs have the same form: a tabular display of data. For example, the **Demog/Groups** tab shows population statistics by group:



| Group | Nbhood | BasePop | CurrPop | Displaced | Attrition |
|-------|--------|---------|---------|-----------|-----------|
| CACB | NAC | 750000 | 750000 | 0 | 0 |
| CACP | NAC | 750000 | 750000 | 0 | 0 |
| CACU | NAC | 1500000 | 1500000 | 0 | 0 |
| CANB | NAN | 500000 | 500000 | 0 | 0 |
| CANP | NAN | 1000000 | 1000000 | 0 | 0 |
| CANU | NAN | 1500000 | 1500000 | 0 | 0 |
| CANV | NAN | 100000 | 100000 | 0 | 0 |
| CASE | NAS | 600000 | 600000 | 0 | 0 |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Demog/Groups

Displays, for each group, the neighborhood in which the group resides, its base population as of time 0, its current population in the neighborhood, the number of people displaced to other neighborhoods, and the number of people killed.

Demog/Nbhoods

Displays, for each neighborhood, whether the neighborhood is part of the local economy and its current population, which is broken down into various categories.

11.13 The Econ Tab

The **Econ** tab has several sub-tabs which display the economics of the playbox in various ways. This tab is shown only in Simulation Mode.

Most of the sub-tabs have the same form: a tabular display of data. For example, the **Econ/Capacity** tab shows production capacity by neighborhood:



| ID | Neighborhood | Urbanization | Prod. Cap. | Cap. Cal. | Cap at T0 | Cap Now |
|-----|--------------|--------------|------------|-----------|-----------|---------|
| NAS | AS | RURAL | 1.00 | 1.026B | 1.026B | 1.026B |
| NAC | AC | RURAL | 1.00 | 1.539B | 1.539B | 1.539B |
| NAN | AN | RURAL | 1.00 | 1.590B | 1.590B | 1.590B |
| NHS | HS | URBAN | 1.00 | 18.281B | 18.281B | 18.281B |
| NHW | HW | URBAN | 1.00 | 4.240B | 4.240B | 4.240B |
| NHE | HE | SUBURBAN | 1.00 | 4.240B | 4.240B | 4.240B |
| NFS | FS | URBAN | 1.00 | 5.240B | 5.240B | 5.240B |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Econ/Overview

Displays an overview of the playbox economy in the form of a spreadsheet.

Econ/Capacity

Displays the production capacity of each neighborhood.

Econ/Population

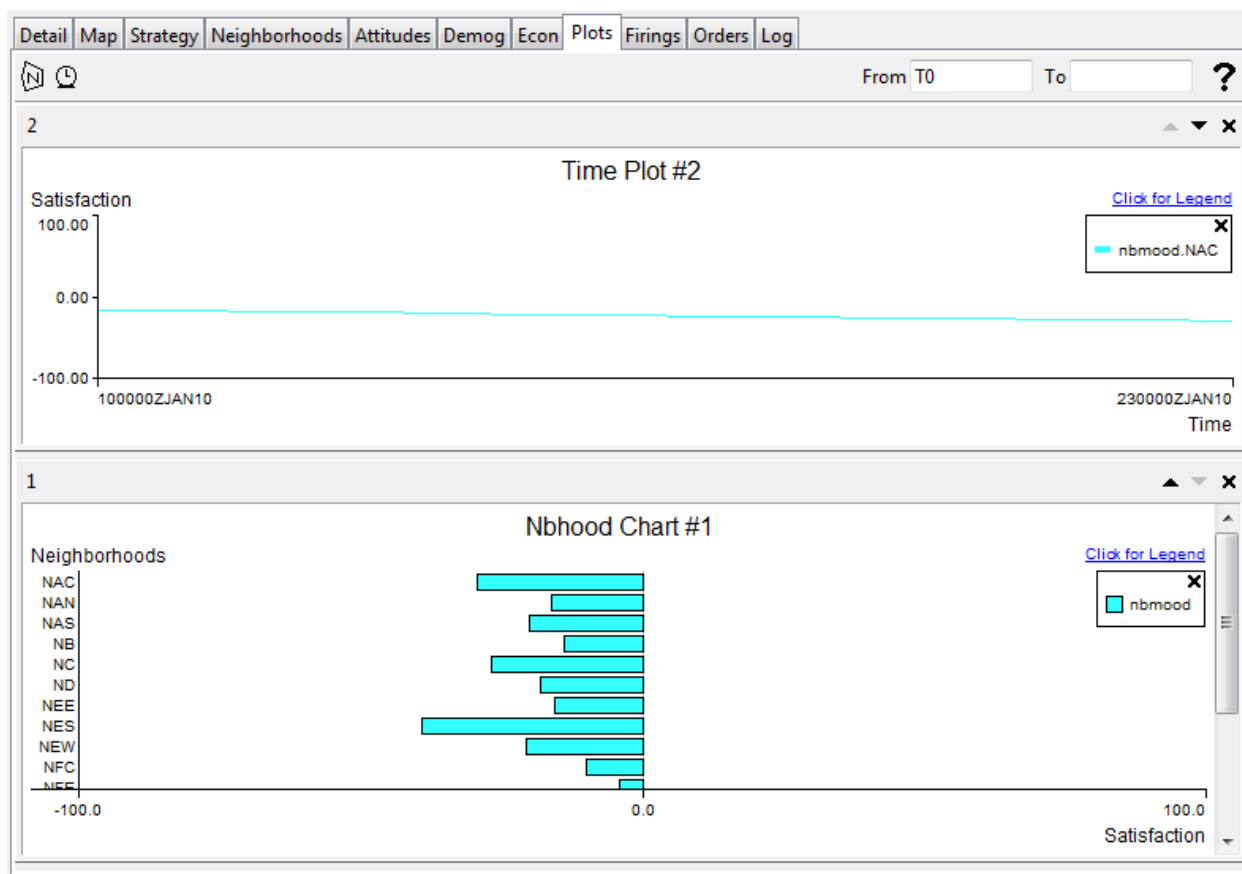
Displays the neighborhood populations broken down into economic categories, along with unemployment statistics.

Econ/Groups

Displays the group populations broken down into economic categories, along with unemployment statistics.

11.14 The Plots Tab

The **Plots** tab, which is displayed only in Simulation Mode, contains the Plot Browser. Here the user can create and browse graphical plots of a variety of simulation variables:






There are two kinds of plot, neighborhood bar charts and time series plots. Each kind of plot can display zero or more display variables, of which there are (again) two kinds: neighborhood variables and time series variables. You can find a complete list of the available display variables in the on-line help; select **Help/Display Variables** from the menu, or simply click the question mark icon on the browser's toolbar.

The Plot Browser will display an arbitrary number of plots in a vertical stack. The border between two plots can be dragged to make one larger and the other smaller. Plots can also be created, destroyed, and moved up and down in the stack.

11.14.1 Plot Browser Toolbar

The Plot Browser's toolbar contains the following controls:

| Control | Description |
|---|---|
|  | New Neighborhood Bar Chart. Creates an empty bar chart. |
|  | New Time Series Plot. Creates an empty time series plot. |
| From <input type="text" value="T0"/> To <input type="text"/> | Time Window. Specifies the end points for the time series plots. The data values are time specification strings. |
|  | Help. Displays the display variable page in the on-line help. |

11.14.2 Plot Toolbar

Each plot appears in a small window embedded within the Plot Browser; and each has the following controls:



The first two controls move the plot's window up or down one slot in the stack; the third closes the plot's window, thus removing it from the stack.

11.14.3 Plot Context Menu

Each plot has a context menu used to add and remove display variables. Right-click on the plot to see the menu. The menu items are as follows:

Add Variable...

Displays a dialog box in which you can enter the name of a display variable. The named variable will be added to the plot.

Remove Variable

Each variable displayed in the plot is listed on this submenu; pick one to remove it from the plot.

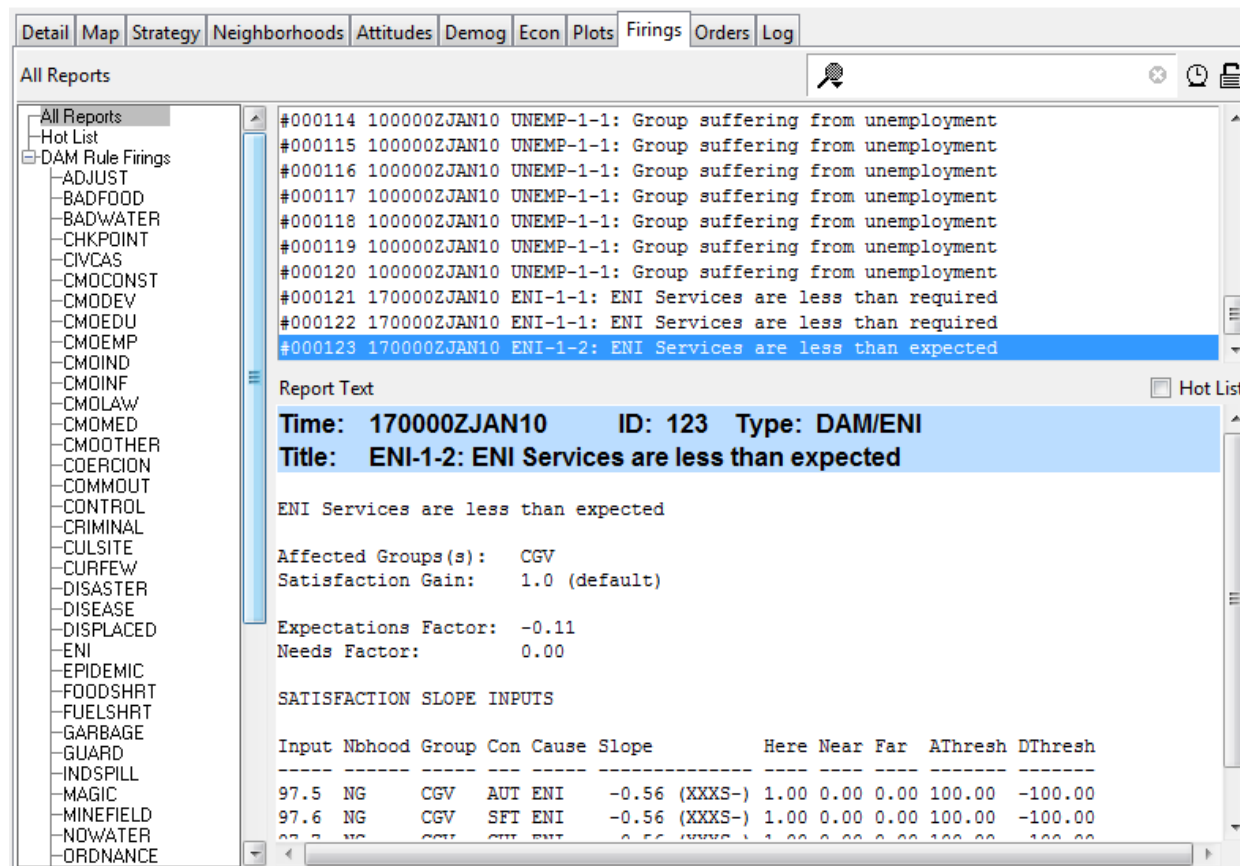
Set Title...

Displays a dialog box in which you can edit the plot's title.

11.15 The Firings Tab

The **Firings** tab contains the Rule Firings browser. This tab is normally not shown; to see it, select **View/Rule Firings** from the menu.

Whenever one of the DAM Rules fires (Section 5.7), information about the rule firing is formatted as a report and sent to the Rule Firings browser:



The browser has four major components:

- The toolbar
- The Bin Tree, which lists bins containing the reports for the different rule sets.
- The Report List, which lists all of the reports in the currently selected bin in time order.
- The Report Text pane, which displays the text of the report currently displayed in the list.

In brief:

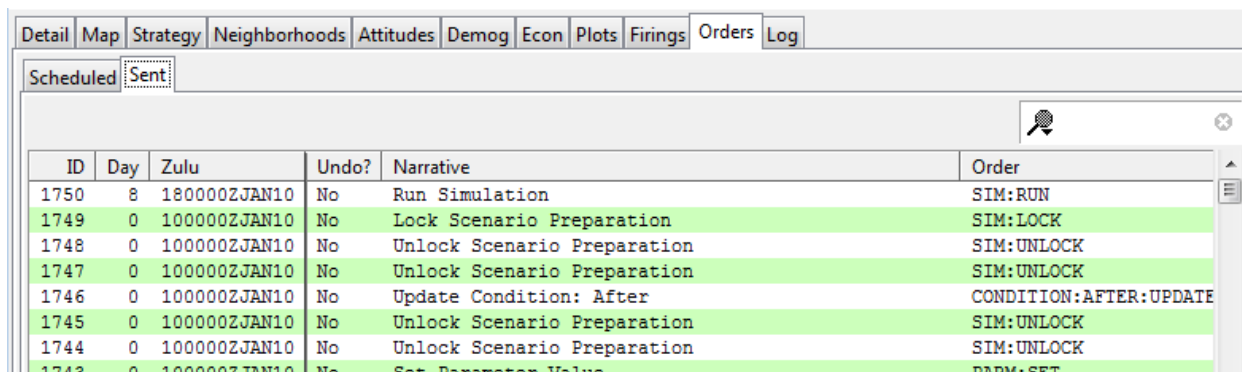
- Select a rule firing from the Reports List to see information about it in the Report Text pane.

- Select a rule set name from the Bin Tree to limit the listed rule firings to rules from that rule set.
- Double-click a rule firing in the Reports List to pop up the rule firing report in a separate window. This allows the user to look at several firings at once.

11.16 The Orders Tab

The **Orders** tab has two sub-tabs which display the simulation orders sent by the user. This tab is usually hidden; to reveal it, select **View/Order History** from the menu.

Both of the sub-tabs have the same form: a tabular display of data. For example, the **Orders/Sent** tab shows all of the orders sent interactively by the user, in reverse order:



The screenshot shows the Athena interface with the 'Orders' tab selected. Below it, the 'Sent' sub-tab is active. A toolbar at the top includes buttons for 'Detail', 'Map', 'Strategy', 'Neighborhoods', 'Attitudes', 'Demog', 'Econ', 'Plots', 'Firings', 'Orders', and 'Log'. The 'Orders/Sent' sub-tab has a search filter box on the right. The main table displays the following data:

| ID | Day | Zulu | Undo? | Narrative | Order |
|------|-----|--------------|-------|-----------------------------|------------------------|
| 1750 | 8 | 180000ZJAN10 | No | Run Simulation | SIM:RUN |
| 1749 | 0 | 100000ZJAN10 | No | Lock Scenario Preparation | SIM:LOCK |
| 1748 | 0 | 100000ZJAN10 | No | Unlock Scenario Preparation | SIM:UNLOCK |
| 1747 | 0 | 100000ZJAN10 | No | Unlock Scenario Preparation | SIM:UNLOCK |
| 1746 | 0 | 100000ZJAN10 | No | Update Condition: After | CONDITION:AFTER:UPDATE |
| 1745 | 0 | 100000ZJAN10 | No | Unlock Scenario Preparation | SIM:UNLOCK |
| 1744 | 0 | 100000ZJAN10 | No | Unlock Scenario Preparation | SIM:UNLOCK |
| 1743 | 0 | 100000ZJAN10 | No | Set Parameter Value | PRM:SET |

Note the filter box at the right-hand end of the toolbar. Type any string into the filter box to display only the matching rows. Click on the “sieve” icon in the filter box to pull down the filter options menu.

The on-line help goes into great detail about the data columns listed on each of the sub-tabs; select **Help/Application Tabs** from the menu, and drill down. This section will give a brief description of each sub-tab.

Orders/Scheduled

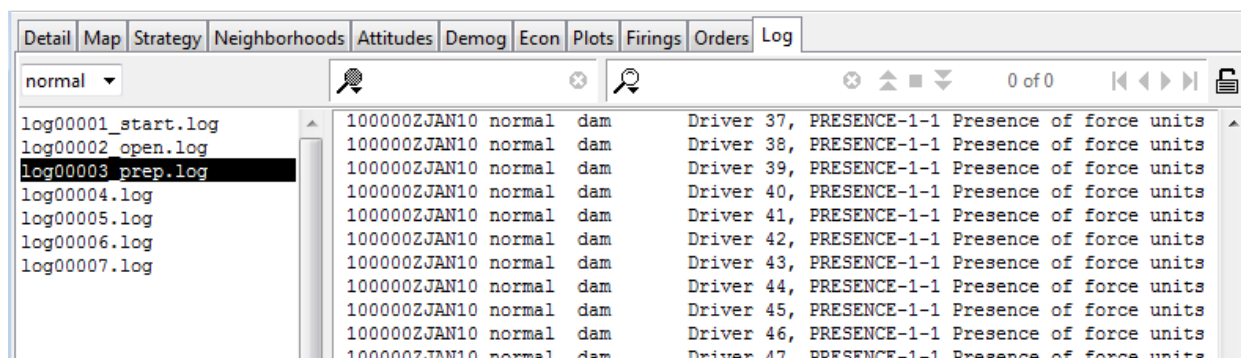
This browser lists all of the orders scheduled in advance by the user and not yet executed. The orders might be scheduled directly by the user, or by an **EXECUTIVE** tactic using the **sched** command.

Orders/Sent

This browser lists all of the orders sent interactively by the user—orders that might, in principle, be undone by selecting **Edit/Undo** from the menu. This set of orders can be exported as an executive script using the **export** command; this is a convenient way to capture the scenario input in script form.

11.17 The Log Tab

The **Log** tab displays the application's debugging log. Because this information is primarily of use to the developer, the **Log** tab is not displayed by default. To view it, select **View/Scrolling Log** from the menu.



The log browser is a powerful tool for searching and filtering logs so as to find the data you need; however, it is rarely needed by the end users. See the on-line help for specifics.

12. CREATING AN ATHENA SCENARIO

This section proceeds step-by-step through the process of creating an Athena scenario. In particular, it is intended to identify all of the things to keep in mind in order to create a high-quality scenario. Section 20 contains this same process in the form of a check-list; it might be useful to print out this check-list when putting together new scenarios.


The steps need not necessarily be done in the order given here; for example, this section has the user create all neighborhoods before creating any groups, whereas one can certainly go back and forth.

Note: All of this work must be done while the scenario is unlocked.

12.1 The Actors

The first step is usually to choose the actors in the playbox. Deciding who the actors are, and which ones you need in the scenario, requires a deep knowledge of the region of interest and is beyond the scope of this document. Here are some things to remember:

- An actor can represent a particular individual, or personify some larger group or entity. In Afghanistan, Hamid Karzai might be an actor. The Afghan government might be an actor. The USA might be an actor.
- Actors are essentially decision makers; they do not, by themselves, occupy neighborhoods or feed the hungry. They need boots (or shoes) on the ground to implement their decision. That is, they need to own force and organization groups who provide them with personnel.

To create an actor, go to the **Groups/Actors** tab, and press the Add Actor button: . Enter the actor's ID and attributes in the resulting dialog. See Section 14.2 or the on-line help for a description of the actor attributes.

The two primary decisions to make here concern the actor's financial resources (which are not yet connected to the Economic model) and the actor's default use of his political support. Does he seek control for himself? Does he work to support some other actor? Or does he remain aloof from the political process?

12.2 The Map

The next step is usually to choose a map of the desired region. It isn't necessary to use a map, as it is used purely for visualization and has no effect on the simulation results; instead, you can draw your neighborhoods on a blank canvas instead. The choice is yours.

If you choose to use one, your map should be stored as a PNG, GIF, or JPEG image file, saved at the resolution you'd usually prefer to look at. Simpler maps are generally more pleasant to use than highly detailed maps.


To use a map, select **File/Import Map ...** from the Athena menu, and select the map file in the resulting dialog box. The imported map is saved in your scenario file; you do not need to keep the map file around once it has been imported.

12.3 The Neighborhoods

Next, define your neighborhoods, remembering that neighborhoods are simply conveniently-sized patches of ground, where “conveniently” may vary from scenario to scenario. If your playbox should encompass a single city, the neighborhoods might be neighborhoods in truth. If your playbox encompasses an entire country, the neighborhoods might be districts or provinces.

The important thing is to put neighborhoods where you need them and not where you don't need them. If you are concerned primarily with the north of the country, you do not need to model the south of the country in as much detail—or, possibly, at all.

In general, Athena is more concerned about neighborhood population than about the geographical size of a neighborhood. However, the size of your neighborhoods will have an effect on how you set the neighborhood proximities (Section 3.2.2).

To define a neighborhood, go to the **Map** tab and click the Create Neighborhood icon: . The **Create Neighborhood** dialog will appear. See Section 14.1 for a description of the neighborhood attributes, or click the question mark icon in the dialog for the on-line help.

To enter the reference point, click in the field on the dialog and then click on the point on the map; the map reference string will appear in the field. Similarly, to enter the polygon click in the Polygon field and click on the map; you will begin drawing the polygon. When the polygon is complete, the vertices will appear in the field.

To redraw a polygon, just repeat the process.

When you have the neighborhood defined as you like, press **Send & Close**.

12.4 Neighborhood Proximities

Athena needs to know how close together or how far apart neighborhoods are, from a social rather than geographic point of view. This is called the *proximity* of one neighborhood to another, and it is defined qualitatively using the values **HERE**, **NEAR**, **FAR**, and **REMOTE**. Note

that it need not be symmetric: down-town might be **NEAR** to the people in the suburb, but the suburb might be **FAR** to the people who live down-town.

Proximity is used in two ways.


First, proximity governs the spread of satisfaction and cooperation effects across the playbox. When civilian casualties occur in neighborhood N1, for example, there may be effects on the people in N2 as well. These effects depend primarily on the horizontal relationship between the pairs of groups, but also on proximity: violence next door affects my sense of safety more than does violence across the country. Thus, indirect effects are highest **HERE**, in the same neighborhood; are lower in **NEAR** neighborhoods, lower still in **FAR** neighborhoods, and negligible in **REMOTE** neighborhoods.¹¹

Second, proximity plays into the Security model (Section 3.8): groups can call on their friends in **NEAR** neighborhoods.

Neighborhood size should affect your proximity settings as follows:

- If your neighborhoods are small, then neighborhood will more often be **NEAR** each other.
- If your neighborhoods are large, then neighborhoods will more often be **FAR** or **REMOTE**; few will be **NEAR**.

In addition, if most of your neighborhoods are large you might wish to set the `force.proximity` model parameter to 0. This parameter determines the extent to which a group is affected by the presence of friendly or enemy groups in **NEAR** neighborhoods.

To set neighborhood proximities, go to the **Neighborhoods/Relationships** tab. By default, every neighborhood is **HERE** to itself (this cannot be changed), and **REMOTE** from all other neighborhoods; to change the proximity, click on a row in the browser, and press the Edit button: .

Note: This dialog also allows you to set the *Effects Delay*. This value allows indirect effects in one neighborhood from an event or situation in another to be delayed by some amount of time. The Effects Delay was originally defined for use in JNEM; typical delays ranged from 0.0 to at most a couple of days. In the Athena context, the Effects Delay adds little value, and it will likely be removed in the future. It is best to leave it set at 0.0.

¹¹ See the values of p and q for each rule set in the *Athena Rules Document* to see how effects are reduced by decreasing proximity for that rule set.

12.5 Civilian Groups

Next, the civilian groups must be defined and added to the neighborhoods. As with actors, the task of how best to divide a region's population into distinct groups requires a deep knowledge of the region and is beyond the scope of this document. However, here are things to keep in mind:

- Belief systems are crucial. A group's belief system gives it its identity. The group's responses and reactions to other groups will depend on its relationships, and its relationships depend on its belief system.
- Each group resides in a single neighborhood. If you wish to have the same group reside in two neighborhoods, create two groups, one in each, and give them similar or identical belief systems.
- Every neighborhood needs at least one civilian group. If you've got no civilian group to put in a neighborhood, delete the neighborhood.

To create a civilian group, go to the **Groups/CivGroups** tab and press the Add Civilian Group button: **+**. See Section 14.3 for a description of a civilian group's attributes.

12.6 Force Groups


Next, the force groups must be defined. Each such group is owned by an actor, and its personnel may be deployed and otherwise made use of by that actor. Remember that force groups do not operate on their own; rather, they do what their owning actor tells them to do. There is a corollary to this: force groups do not have belief systems of their own, but instead each force group inherits the beliefs and affinities of its owning actor.

It is up to the analyst to know the important force groups currently operating in the playbox.

To define a force group, go to the **Groups/FrcGroups** tab, and press the Add Force Group button: **+**. The most important decisions to make with regard to force groups are the force type, the deployment and attack costs, and whether or not the group is uniformed. See Section 14.4 for a description of a force group's attributes.

12.7 Organization Groups

Like force groups, organization groups are assets belonging to actors. The primary differences between the two are that organization groups do not carry out military operations, and do not affect the civilians by their mere presence.

To define a force group, go to the **Groups/OrgGroups** tab, and press the Add Organization Group button: . See Section 14.5 for a description of an organization group's attributes. The most important decisions to make with regard to organization groups are the organization type and the deployment costs.

Every organization group must belong to some actor. Humanitarian relief organizations are often apolitical (at least so far as the playbox is concerned); if there is no appropriate actor present in the playbox, then define a dummy actor, indicate that it supports no actor politically, and use its strategy to control such apolitical organizations. In such a case, the group's deployment cost per person per week can usually be set to zero.


12.8 Belief Systems

Next, belief systems must be defined for each of the actors and civilian groups. This can be a complex (and enlightening) process. To build a belief system, go to the Belief System browser on the **Groups/Beliefs** tab. The subsequent steps are described in the following subsections.

12.8.1 Define the Topics

First, consider the topics on which the residents of the playbox have strongly held beliefs. Some of these will be basic to the culture or cultures in the playbox (e.g., beliefs about the role of women) and some will be more topical (e.g., beliefs about the presence of U.S. forces in the region). It is useful to identify the cultural and political fault lines in the playbox, as these will often reveal the topics on which groups disagree.

Topics must be stated in absolute terms, so as to mean the same thing to all groups. "Public safety" might be a topic, but if so it must be understood as the safety of the public in general, not the safety merely of "my group".

Topics are defined by pressing the "Add Topic" button in the top half of the Belief Systems browser, , and filling out the resulting dialog.

12.8.2 Define the Beliefs

Then, the actors and civilian groups are shown in the "Entities" tree in the lower left part of the Belief Systems Browser. Select an actor or group to edit its beliefs.

Each belief consists of the selected entity's position on a topic, a qualitative variable that ranges from Passionately For to Passionately Against, and an emphasis on agreement or disagreement with respect to that topic. A group that emphasizes disagreement strongly will be more likely to have strongly negative affinities with those groups that disagree with it on the topic.

12.8.3 Compute the Affinities

Once you have defined some or all of the beliefs, it is time to compute and examine the affinities. The lower right corner of the Belief Systems Browser shows the affinity of the selected entity with all other entities, and their affinities to him. Each affinity is expressed as a number from -1 to +1.

By default, affinities are computed automatically, whenever a topic is added or a belief is changed. For large scenarios with many actors and civilian groups, this can be quite slow. In this case, de-select the **Auto-Calc** checkbox, which is toward the top of the Belief Systems Browser; then, press the **Calc Now** button when you wish to compute the affinities.

12.8.4 Adjust the Affinities


It is likely that the affinities will not look quite right at first. In this case, there are a number of things to try:

- If two groups known to be enemies have a positive affinity, look for additional topics that divide them and add them in.
- If two groups known to be friends have a negative affinity, look for additional topics on which they agree and add them in.
- Use the **Playbox Commonality** slider to adjust the degree of cultural commonality in the playbox. Increasing the playbox commonality is like adding topics of general agreement, and will tend to make all affinities increase; decreasing the playbox commonality is like removing such topics and will tend to make all affinities decrease.
- Use the **Entity Commonality** slider to decrease a particular entity's participation in the common culture of the playbox; this is especially useful for foreign actors (e.g., the U.S.A.). Reducing the entity's Entity Commonality is like reducing the entity's agreement with the residents of the playbox on basic cultural topics, and will tend to decrease the entity's affinity with other entities.

12.9 Horizontal Relationships

Horizontal relationships are the relationships between groups, based on their affinities for each other; their affinities derive from their belief systems, where force and organization groups are assumed to inherit the beliefs of their owning actors. The horizontal relationships produced in this way can be seen on the **Groups/Relationships** tab.

In general, relationships should be adjusted by tweaking the belief systems of the relevant groups. Alternatively, the analyst may choose to override the affinity-based relationship with some preferred value. To do this, select the relevant relationship on the


Groups/Relationships tab and press the Edit button, . Enter the desired value in the dialog box.

12.10 Group Satisfaction

Next, it is necessary to state the initial satisfaction levels and saliencies for each concern of each civilian group. The four concerns are defined in Section 5.4.

By default, a civilian group is ambivalent (neither satisfied nor dissatisfied) about all four concerns, and gives all four the same importance (the saliency of each concern is 1.0). Neither of these defaults is realistic. Some set of circumstances obtains at time 0, and the civilians will have attitudes about it; and groups really do give more or less importance to different things. For example, a group from an honor-based culture might give Culture (CUL) a much higher saliency than Safety (SFT) or Quality of Life (QOL). Saliency is important because the group's mood is their average satisfaction across their concerns as weighted by the saliency.


The second thing to consider is the trends and thresholds for each satisfaction curve, which can be used to implement a kind of regression to a natural value for the satisfaction level.

To edit a group's satisfaction level, go to the **Attitudes/Satisfaction** tab and select the concern in the browser. Press the Edit button, , and enter the desired changes in the resulting dialog. See Section 14.6 for a detailed description of the attributes of a satisfaction level.

12.11 Group Cooperation

Next, it is necessary to consider the initial levels of cooperation between the civilian groups and the force groups. Remember that cooperation has to do with the flow of intelligence between the groups; it can be thought of as the likelihood, 0% to 100% that a member of the civilian group will give intelligence information to a member of the force group. Cooperation defaults to 50%, which is a reasonable choice lacking other information, but an analyst with a knowledge of the groups in the playbox should do better.

Like satisfaction curves, cooperation curves have trends and thresholds; these can be used to make the curve regress back to some natural level in the absence of other drivers.


To edit a cooperation level, go to the **Attitudes/Cooperation** tab and select the cooperation curve in the browser. Press the Edit button, , and enter the desired changes in the resulting dialog. See Section 14.7 for a detailed description of the attributes of a cooperation level.

12.12 Status Quo Deployments

The next essential step is to determine the status quo deployment of force and organization group personnel across the neighborhoods of the playbox. This status quo deployment has two effects: it determines the state of affairs prior to the start of the simulation, and it determines the number of personnel initially available for deployment once the simulation starts.


While the simulation is running, the deployment of a group's personnel is determined by the strategy of the group's owning actor. At time 0, the actor's strategy is conditioned by the state of affairs in the playbox prior to that time—and that state of affairs depends to a considerable extent on the deployment of personnel. Thus, Athena needs to know what that deployment is in order for the initial strategy execution to make sense.

In addition, if a group has no status quo deployment then there will be no troops for the actor deploy unless he explicitly brings them into the playbox using the **MOBILIZE** tactic.

To set a group's status quo deployments, go to the **Groups/Deployments** tab and select a neighborhood/group pair in the browser. Press the Edit button, , and enter the desired number of personnel in the resulting dialog.

12.13 Status Quo ENI Funding


Status quo ENI funding is like the status quo deployments: we need to know how much money is being spent on Essential Non-Infrastructure (ENI) services for each group prior to the start of the simulation, because this determines the state of affairs in which the initial strategy execution will take place; it also determines each group's initial expectations regarding the provision of ENI services.

To set the status quo level of funding for ENI services for a civilian group by an actor, go to the **Groups/Services** tab and select the Group/Actor pair. Press the Edit button, , and enter the desired level of funding in the resulting dialog.

Practically speaking, it may be necessary to iterate this step, locking the scenario, looking at the group's expected level of service in the Detail browser, unlocking the scenario, and updating the status quo funding.

12.14 Environmental Situations

Environmental situations represent on-going problems in neighborhoods, e.g., an epidemic in this neighborhood or a power outage in that neighborhood. If any such exist at the start of the simulation, they should generally be created at this time.

To create an environmental situation, go the **Map** tab, and press the Create Environmental Situation button: . See Section 14.8 for a description of the attributes of environmental situations.

12.15 Strategies

The final step is to define a strategy for each actor. The subtleties of defining a realistic strategy are beyond the scope of this section; instead, it will give an overview of the process and note a few general principles. See Sections 15 and 16 for documentation of the available tactics and conditions, and Part III of this document for specifics of how to accomplish particular things.

Strategies are edited in the Strategy Browser, which is found on the **Strategy** tab. Select the actor from the list on the left, and then add goals, tactics, and conditions using the toolbar buttons.

12.15.1 The Role of an Actor's Strategy

An actor's strategy has two roles. First, the actor's strategy models his decision-making process: it specifies which tactics the actor will choose to execute under varying circumstances. This is the way we usually think of strategies.

But second, and less obviously, it can also model simulation events that directly affect the actor and his assets that are beyond the actor's control. For example, suppose the analyst believes that Actor A's troops will desert him if he loses control of the neighborhood that represents the capital city. This can be handled straightforwardly: Actor A's strategy can demobilize his troops (using the **DEMOB** tactic) if the **CONTROL** condition indicates that he no longer controls the relevant neighborhood. This is something that happens to Actor A rather than something that Actor A chooses, but his strategy can make it happen.

12.15.2 Order Matters

The order in which tactics are listed in the actor's strategy matters greatly, because he will attempt to execute them in that order. If assets are consumed before the end of the list is reached, lower priority tactics might not be executed. If tactics near the top of the list change current circumstances, the conditions on lower priority tactics might not be met.

12.15.3 Use of Goals

It is possible to use the Goals part of the Strategy browser simply to record the actor's goals while pondering his strategy. Practically speaking, though, many of an actor's real world goals will not translate directly (or, in some cases, at all) into Athena terms.

A more pragmatic use of goals is to treat them as *ad hoc* named conditions. If you find that you are repeating a particular set of conditions on multiple tactics, consider defining a new goal and attach the set of conditions to it. You can then attach the **MET** or **UNMET** condition to the tactics in place of the full set of conditions.

Used this way, a goal can represent two things: a state of affairs that the actor wishes to bring about, or a state of affairs that the actor wishes to preserve. In the former case, the actor will execute tactics when the goal is **UNMET**; in the latter when the goal is **MET**.

12.16 The Model Parameter Database

Athena defines a vast number of user-customizable parameters. These *model parameters* are used to calibrate models, to enable or disable features, and (in a few rare cases) as scenario inputs. Although it is not usually necessary for analysts to change these parameters, it is often useful to examine their values; for example, the model descriptions in the *Athena Analyst's Guide* frequently refer to model parameters by name. The complete set of model parameters is referred to as the *model parameter database*.

This section gives a general overview of the model parameter database and its contents, as well as how to learn more about it, browse its contents, and even make changes.

12.16.1 Model Parameters

A model parameter is essentially a named variable with a specific data type and a default value. For example, the `force.proximity` parameter determines the extent to which nearby neighborhoods contribute to a group's force in a neighborhood, which in turn affects security and volatility. Its value must be a number no less than 0.0; its default value is 0.1.

Parameter names are hierarchical. For example, all parameters used by the force and security model have names beginning with "`force.`"; we will sometimes refer to the whole set as the "`force.*`" parameters.

The naming hierarchy can have many levels. The "`dam.BADFOOD.nearFactor`" parameter, for instance, gives the "near factor", or "p" value for the Driver Assessment Model's **BADFOOD** rule set. The parameters that apply to this rule set all have names like "`dam.BADFOOD.*`".

12.16.2 Browsing the Model Parameter Database

There are two ways to browse the contents of the model parameter database. The first is to go to the Detail Browser, and select "Model Parameters" or one of its children from the Objects tree in the sidebar. This allows you to see either all model parameters or one of the top-level

subsets (e.g., "force . *"); for each parameter you will see its name, its default value, and its current value. If the current value differs from the default, it will be shown in red.

The parameter names link to the on-line help; click on a name to read more about the parameter.


The second approach provides the same information for a narrower subset of parameters. To see only those parameters related to the **BADFOOD** rule set, for example, select **Reports/Model Parameters Report...** from the menu and enter "dam . BADFOOD . *" in the "Wildcard" field.


12.16.3 Parameter Help

The on-line help includes documentation on each model parameter, arranged hierarchically by parameter name. There are many ways to access this information:

- Select **Help/Model Parameters** from the menu
- Select "Model Parameters" or one of its children from the "?" tab in the Detail Browser's sidebar.
- Click on a model parameter name when browsing model parameters in the Detail Browser.

12.16.4 Editing Model Parameters

Before editing a model parameter, make sure you know what it is for and what the range of acceptable values is. Then, there are two ways to edit it. The usual way is to browse the model parameters in the Detail Browser; each parameter name has associated with it a pencil icon, . Click on the icon, and enter the new value in the resulting dialog box.

Some parameters can only be edited when the scenario is unlocked; if the scenario is locked, the pencil will be replaced by a padlock icon, . Note, however, that it is almost never a good idea to edit model parameters when the scenario is locked, except when running tests or troubleshooting a problem, because the parameter changes will be lost when the scenario is unlocked.

It is also possible to edit model parameters from the Athena command line or via an executive script using the **parm set** command. See the on-line help for more information.

12.16.5 Model Parameter Files

It is sometimes desirable to copy model parameter settings from one scenario to another. For example, an analyst might calibrate Athena for a particular region or use, and wish to use these

settings for a number of scenarios. Consequently, Athena allows the user to import and export the model parameter database.

To export the model parameter database, select **File/Parameters/Export...** from the menu; you will be prompted to enter the name of a “.parmdb” file. To import the saved parameters into another scenario, open the scenario and select **File/Parameters/Import...** from the menu and select the “.parmdb” file.

Part III: Athena Cookbook

13. RECIPES

This section contains procedures for accomplishing particular tasks or effects within an Athena scenario.

13.1 Disabling the Economic Model

Athena's three-sector CGE is essentially a prototype and a proof-of-concept; while it is not without value, it is limited in its applicability. We have designed a more robust six-sector CGE, which we expect to implement in Athena 4. For this reason, users might wish to disable the Economic model, as shown here:

- 1) Unlock the scenario, if it is locked.
- 2) In the Detail Browser, go to the sidebar's **Objects** tab and click on the "econ.*" entry under "Model Parameters".
- 3) The resulting page will list the `econ.disable` parameter at the top.
- 4) Click on the pencil icon to the right of the parameter.
- 5) Enter "yes" in the "Value:" field of the resulting dialog, and press "Send&Close".

The Economic model is now disabled. Set the parameter back to "no" to enable it again.

13.2 Calibrating the Economic Model

This section describes how to calibrate Athena's Economic Model it for a particular region.

- 1) Unlock the scenario, if it is locked.
- 2) Make sure that the `econ.disable` model parameter is set to "no".
- 3) Calibrate the remaining econ.* model parameters as described in the *Athena Analyst's Guide*.
- 4) Lock the scenario, and verify that there are no sanity check failures.

13.3 Regression of an Attitude to a Mean

By default, satisfaction and cooperation levels persist. That is, if the attitude drivers to date have moved a group's SFT satisfaction to -40, and if there are no active attitude drivers, the group's SFT satisfaction will remain at -40 indefinitely.

This is not particularly realistic. In general, we would expect SFT satisfaction to decrease when civilian casualties occur, but to increase slowly in between. It is possible to do this using the trends and thresholds on the relevant curve.

Suppose, for example, that we would like group G's SFT satisfaction to increase to +40 in the absence of other drivers. We would set this up as follows:

- 1) Go to the **Attitudes/Satisfaction** tab.
- 2) Select group G's SFT satisfaction curve.
- 3) Set the Ascending Trend to some positive value, say, 1.0 points per day
- 4) Set the Ascending Threshold to +40

With these settings SFT will increase by 1.0 point per day so long as SFT is less than +40; once it reaches +40, the effect will stop.

Similarly, the Descending Trend and Threshold can be used to drive attitudes down. Suppose G's QOL satisfaction should always return to 0.0 in the absence of other drivers:

- 1) Set the Ascending Trend to 1.0
- 2) Set the Descending Trend to -1.0
- 3) Set both thresholds to 0.0

While the QOL level is above zero, it will go down by one point per day; while it is below zero, it will go up by one point per day.

The same techniques can be used for cooperation levels.

Part IV: Reference

14. ATHENA OBJECTS

This section gives a complete description of the data attributes associated with the major Athena objects: neighborhoods, actors, and groups. This information is also available in the on-line help; however, the emphasis there is on helping the user to enter valid data. The emphasis here is on how the attributes are used by Athena.

Each attribute has two names: the human-readable name used in dialog boxes, browsers, and reports, and the name used in SQLite database tables.

14.1 Neighborhoods

A neighborhood is a polygonal area on the map in which simulation events occur. Civilian groups live in neighborhoods; force groups operate in neighborhoods; actors attempt to control neighborhoods.

The following table shows the attributes associated directly with each neighborhood. In addition to these, each neighborhood has a proximity to every other neighborhood; and of course, many, many things are explicitly related to particular neighborhoods.

| Attribute | Description |
|------------------------------------|---|
| Neighborhood ID (n) | The neighborhood ID, a unique string used to identify this neighborhood. It should be short, and may consist of capital letters and digits. It must be distinct from all other entity IDs used in the scenario (e.g., group IDs). |
| Long Name (longname) | A longer, human-readable name for use in reports and other output; it may also be used to document the precise meaning of the neighborhood ID, which might otherwise be cryptic. |
| Local Neighborhood? (local) | A flag that indicates whether or not this neighborhood is part of the local economy described in Section 7. Local neighborhoods contribute consumers, workers, and production capacity to the local economy; non-local neighborhoods do not. |
| Urbanization (urbanization) | The urbanization of the neighborhood: one of Urban , Suburban , Rural , or Isolated . Urbanization affects the attrition model (Section 3.12) and the ENI Services model (Section 3.13). |
| Controller (controller) | The ID of the actor in control of this neighborhood prior to the start of the simulation (if any). As the simulation runs, the actor in control of the neighborhood may change. |
| Volatility Gain (vtygain) | A gain control on the volatility of the neighborhood, nominally set to 1.0. Increasing or decreasing the gain increases or decreases the volatility proportionally; this in turn will increase or decrease the security of the groups in the neighborhood. Thus, this control can be used to tweak the security of neighborhoods that are known to be more or less peaceful than one would expect. |
| Polygon (polygon) | The neighborhood's border, expressed as a list of map coordinates. |
| Reference Point (refpoint) | The neighborhood's reference point—a point that is known to be within the neighborhood. The software uses this point to detect improperly overlapping or nested neighborhoods, and also as a safe place to position icons relative to the polygon. |
| Stacking Order (stacking_order) | Overlapping neighborhood polygons stack like elements in a drawing program. When a point on the map is contained in two or more neighborhood polygons, it is presumed to be in the neighborhood at the top of the stacking order. |
| Obscured By (obscured_by) | It's an error for a neighborhood higher in the stacking order to cover a lower neighborhood's reference point. If this should happen to a neighborhood during scenario development, this attribute will be set to the ID of the offending neighborhood. |

14.2 Actors

Actors are the significant decision makers in the playbox. Actors have belief systems, own force and organization groups, and own and execute strategies, which consist of goals, tactics, and conditions; see Sections 15 and 16 for information on the different tactic and condition types.

The following data attributes are directly associated with each actor.

| Attribute | Description |
|--------------------------------|--|
| Actor ID (a) | The actor ID, a unique string used to identify this actor. It should be short, and may consist of capital letters and digits. It must be distinct from all other entity IDs used in the scenario (e.g., group IDs). |
| Long Name (longname) | A longer, human-readable name for use in reports and other output; it may also be used to document the precise meaning of the actor ID, which might otherwise be cryptic. |
| Supports (supports) | <p>This attribute identifies the actor that usually receives this actor's political support (Section 6.2.1); it may be an actor's ID, or SELF if the actor usually supports himself, or NONE if the actor doesn't usually make use of his political support.</p> <p>Political support is computed weekly, neighborhood by neighborhood, and by default actors will support each other as determined by this attribute. However, it is also possible for the actor to override this setting in particular neighborhoods using the SUPPORT tactic (Section 15.13); thus, he can support himself in one neighborhood, a different actor in another neighborhood, and no one at all in a third.</p> |
| Cash Reserve (cash_reserve) | The actor's cash reserve as of the start of the simulation, in dollars; see Section 6.1.1. |
| Income (income) | <p>The actor's regular income¹², in dollars/week. Income flows into the actor's cash-on-hand, where it can be used to fund tactics.</p> <p>Note that the actor can also receive income from other actors, by way of the FUND tactic.</p> |
| Cash-On-Hand (cash_on_hand) | <p>An actor's cash-on-hand is the amount of money the actor has available to fund tactics. The actor receives income into his cash-on-hand each week, and he may also move money between his cash-on-hand and his cash reserve. Cash-on-hand that is unspent during one week rolls over to the next.</p> <p>This attribute represents the actor's cash-on-hand at the start of the simulation: the actor's income, carry-over from the previous week, and any funds received from other actors prior to simulation start.</p> |

¹² At present, this is simply a scenario input. In a future version of Athena, actor's income will flow out of the Economic model and may increase and decrease with the economy.

14.3 Civilian Groups

Civilian groups represent the population of the playbox, broken down by neighborhood, ethnicity, clan membership, political views, or any other criteria the analyst chooses. The identity of a civilian group is determined by its belief system, which determines its affinity with actors and other groups.

The following attributes are directly associated with civilian groups.

| Attribute | Description |
|--|--|
| Group ID (g) | The group ID, a unique string used to identify this group. It should be short, and may consist of capital letters and digits. It must be distinct from all other entity IDs used in the scenario (e.g., actor IDs). |
| Long Name (longname) | A longer, human-readable name for use in reports and other output; it may also be used to document the precise meaning of the group ID, which might otherwise be cryptic. |
| Neighborhood (n) | This attribute is the ID of the group's neighborhood of residence. |
| Demeanor (demeanor) | The demeanor of the group, i.e., its propensity for violence. The group may be aggressive, average, or apathetic. Demeanor affects the group's ability to project force in the Security model (Section 3.8). It has had other uses in the past (i.e., in JNEM) and will likely be used in other models in the future. |
| Base Population (basepop) | The population of this group in its neighborhood of residence just prior to the start of the simulation. This population can be reduced by attrition and by displacement to other neighborhoods. Population size is one of the more important inputs, as it plays a role in many different models. See the <i>Athena Analyst's Guide</i> for details. |
| Subsistence Agriculture Percentage (sap) | The percentage of the group's population that supports itself by subsistence agriculture. The subsistence population does not participate in the cash economy as either consumers or workers. When members of the subsistence population are displaced from their homes, they become consumers, and may become workers as well. |
| Color (color) | Athena represents group personnel on the map using icons. This is the color of this group's icons. It can be entered as a hexadecimal RGB string, e.g., red is "#FF0000"; however, it will usually be chosen using the GUI. The color has no effect on the model results. |
| Unit Shape (shape) | The shape of the group's units: friend, enemy, or neutral, as defined in MIL-STD-2525a. The shape has no effect on the model results. |

14.4 Force Groups

A force group represents a body of personnel that exists primarily to apply armed force at the behest of an actor, though the actor can use them for other purposes as well. The following attributes are directly associated with force groups.

| Attribute | Description |
|---------------------------------|--|
| Group ID (g) | The group ID, a unique string used to identify this group. It should be short, and may consist of capital letters and digits. It must be distinct from all other entity IDs used in the scenario (e.g., actor IDs). |
| Long Name (longname) | A longer, human-readable name for use in reports and other output; it may also be used to document the precise meaning of the group ID, which might otherwise be cryptic. |
| Owning Actor (a) | The ID of the actor that owns this group. This attribute may initially be left blank, but must be filled in before the scenario is locked. |
| Force Type (forcetype) | <p>The kind of force this is: Regular Military, Paramilitary, Police, Irregular Military, and Organized Crime. The force type affects the group's ability to project force in the Security model (Section 3.8).</p> <p>Regular Military includes traditional uniformed armies. Paramilitary includes national police forces trained in military tactics. Irregular Military includes militias and terrorist groups; many of these will be non-uniformed.</p> |
| Demeanor (demeanor) | The demeanor of the group, i.e., its propensity for violence. The group may be aggressive, average, or apathetic. This also affects the group's ability to project force in the Security model (Section 3.8). |
| Cost \$/person/week (cost) | <p>The cost to the owning actor of maintaining one member of the group in the playbox for one week, in dollars. The cost is incurred during deployment each week, hence this is sometimes called the deployment cost.</p> <p>It is reasonable to set this attribute to zero if the number of troops available does not depend on the actor's financial assets in the playbox. A U.S. commander, for example, is given some number of troops; his ability to deploy them is not (to a first approximation, at least) limited by his own budget.</p> |
| Cost \$/attack (attack_cost) | The cost of one attack on another force group, as described in Section 3.12, in dollars. |
| Uniformed? (uniformed) | A flag indicating whether members of this force group are uniformed (and hence easily recognizable) or non-uniformed, blending in with the local population. In the attrition model (Section 3.12), uniformed forces may attack non-uniformed forces, and vice-versa. |
| Color (color) | Athena represents group personnel on the map using icons. This is the color of this group's icons. It can be entered as a hexadecimal RGB string, e.g., red is "#FF0000"; however, it will usually be chosen using the GUI. The color has no effect on the model results. |
| Unit Shape (shape) | The shape of the group's units: friend, enemy, or neutral, as defined in MIL-STD-2525a. The shape has no effect on the model results. |

14.5 Organization Groups

An organization group represents a body of personnel that exists primarily for non-military reasons. Organization groups are owned by actors. The following attributes are directly associated with organization groups.

| Attribute | Description |
|--------------------------------|---|
| Group ID (g) | The group ID, a unique string used to identify this group. It should be short, and may consist of capital letters and digits. It must be distinct from all other entity IDs used in the scenario (e.g., actor IDs). |
| Long Name (longname) | A longer, human-readable name for use in reports and other output; it may also be used to document the precise meaning of the group ID, which might otherwise be cryptic. |
| Owning Actor (a) | The ID of the actor that owns this group. This attribute may initially be left blank, but must be filled in before the scenario is locked. |
| Organization Type (orgtype) | The kind of organization this is: Non-Governmental Organizations (NGOs), like the Red Cross or Doctors Without Borders; Inter-Governmental Organizations (IGOs) like UNESCO; and contractors, like Haliburton. The primary difference between the three types is that contractors can project a small amount of force in the Security model (Section 3.8), and consequently have reduced security requirements when performing activities in neighborhoods. |
| Demeanor (demeanor) | The demeanor of the group, i.e., its propensity for violence. The group may be aggressive, average, or apathetic. This also affects the group's ability to project force in the Security model (Section 3.8). |
| Cost \$/person/week (cost) | <p>The cost to the owning actor of maintaining one member of the group in the playbox for one week, in dollars. The cost is incurred during deployment each week, hence this is sometimes called the deployment cost.</p> <p>It is reasonable to set this attribute to zero if the number of personnel available does not depend on the actor's financial assets in the playbox. Foreign NGOs, for example, might belong to a "pseudo-actor", a dummy actor whose strategy determines what the NGO does. In this case, the primary asset is the people on the ground, and the monetary cost can be ignored.</p> |
| Color (color) | Athena represents group personnel on the map using icons. This is the color of this group's icons. It can be entered as a hexadecimal RGB string, e.g., red is "#FF0000"; however, it will usually be chosen using the GUI. The color has no effect on the model results. |
| Unit Shape (shape) | The shape of the group's units: friend, enemy, or neutral, as defined in MIL-STD-2525a. The shape has no effect on the model results. |

14.6 Satisfaction Levels

Athena defines satisfaction levels for every civilian group and each of the four concerns: Autonomy (AUT), Culture (CUL), Quality of Life (QOL) and Safety (SFT). Each curve has the following attributes, to be set as part of scenario preparation.

The initial satisfaction level and the saliency are the two most important attributes. The trends and thresholds can be used to implement regression to some base value; see Section 13.3.

| Attribute | Description |
|-----------------------------------|--|
| Sat at T0 (sat0) | The group's satisfaction with this concern at time 0, also called the <i>initial satisfaction</i> . This needs to be set for each curve by someone who is acquainted with the groups and circumstances in the playbox. It is frequently useful to use the following scale: Very Satisfied, Satisfied, Ambivalent, Dissatisfied, Very Dissatisfied . |
| Saliency (saliency) | <p>The saliency is the importance of this concern to this group: the greater the saliency relative to the other concerns, the greater the effect this concern has on the group's overall mood. Use the following scale: Crucial, Very Important, Important, Less Important, Unimportant, Negligible.</p> <p>Thus, if personal safety and comfort are less important to the group than honor, the CUL saliency might be Crucial while the QOL and SFT saliences are Less Important.</p> |
| Ascending Trend (atrend) | <p>The ascending trend for this curve, a non-negative slope in nominal satisfaction points per day. This trend is applied to the curve on each day that the current level is less than the Ascending Threshold, that is, it will tend to move the curve up to the Ascending Threshold day-by-day, and stop when it gets there.</p> <p>This trend defaults to 0.0, in which case it has no effect.</p> |
| Ascending Threshold (athresh) | <p>The ascending threshold for this curve, a satisfaction value from -100 to +100. The ascending trend has effect only when the current level is below this threshold.</p> |
| Descending Trend (dtrend) | <p>The descending trend for this curve, a non-positive slope in nominal satisfaction points per day. This trend is applied to the curve on each day that the current level is greater than the Descending Threshold, that is, it will tend to move the curve down to the Descending Threshold day-by-day, and stop when it gets there.</p> <p>This trend defaults to 0.0, in which case it has no effect.</p> |
| Descending Threshold (dthresh) | <p>The descending threshold for this curve, a satisfaction value from -100 to +100. The descending trend has effect only when the current level is above this threshold.</p> |

14.7 Cooperation Levels

Athena defines a cooperation level for every civilian group with every force group. Remember that cooperation is the likelihood that members of the civilian group will give intel to members of the force group when asked; a high cooperation level does not imply by itself any particular degree of assistance beyond the simple giving of information.

Each curve has the following attributes, to be set as part of scenario preparation. The initial cooperation level is the most important attribute. The trends and thresholds can be used to implement regression to some base value; see Section 13.3.

| Attribute | Description |
|-----------------------------------|---|
| Cooperation at T0 (coop0) | The civilian group's cooperation with the force group at time 0, also called the <i>initial cooperation</i> . This needs to be set for each curve by someone who is acquainted with the groups and circumstances in the playbox. It is frequently useful to use the following scale: Always Cooperative, Very Cooperative, Cooperative, Marginally Cooperative, Uncooperative, Very Uncooperative, Never Cooperative . |
| Ascending Trend (atrend) | The ascending trend for this curve, a non-negative slope in nominal cooperation points per day. This trend is applied to the curve on each day that the current level is less than the Ascending Threshold, that is, it will tend to move the curve up to the Ascending Threshold day-by-day, and stop when it gets there. This trend defaults to 0.0, in which case it has no effect. |
| Ascending Threshold (athresh) | The ascending threshold for this curve, a cooperation value from 0 to 100. The ascending trend has effect only when the current level is below this threshold. |
| Descending Trend (dtrend) | The descending trend for this curve, a non-positive slope in nominal cooperation points per day. This trend is applied to the curve on each day that the current level is greater than the Descending Threshold, that is, it will tend to move the curve down to the Descending Threshold day-by-day, and stop when it gets there. This trend defaults to 0.0, in which case it has no effect. |
| Descending Threshold (dthresh) | The descending threshold for this curve, a cooperation value from 0 to 100. The descending trend has effect only when the current level is above this threshold. |

14.8 Environmental Situations

Environmental situations represent problems in neighborhoods that actors can ignore or resolve. The environmental situation model was brought into Athena from JNEM and is not yet completely integrated with the actor/strategy model.

14.8.1 Environmental Situation Types

Athena defines the following environmental situation types:

BADFOOD (Contaminated food supply): The food supply in the neighborhood has been contaminated, leading to poisoning, sickness, and shortages. A **BADFOOD** situation will spawn a **DISEASE** situation after 1 day, and will be automatically resolved in 10 days.

BADWATER (Contaminated water supply): The water supply in the neighborhood has been contaminated, leading to poisoning, sickness, and a shortage of water. A **BADWATER** situation will spawn a **DISEASE** situation after 1 day, and will be automatically resolved in 5 days.

COMMOUT (Communications outage): There is a communications outage (radio, TV, telephones, etc.) in the neighborhood. The situation will be automatically resolved in 7 days.

CULSITE (Damage to cultural site): An important cultural site has been damaged, leading to unrest. The situation will be automatically resolved in 45 days.

DISASTER (Disaster): There has been some kind of disaster in the neighborhood. The situation will be automatically resolved in 45 days.

DISEASE (Disease): There is disease due to environmental conditions in the neighborhood (i.e., disease caused by poor sanitation, contaminated water, etc.). The situation will be automatically resolved in 30 days.

EPIDEMIC (Epidemic): There is epidemic disease in the neighborhood. Note that Athena will not automatically cause the epidemic to spread to other neighborhoods. The situation will be automatically resolved in 360 days.

FOODSHRT (Food shortage): There is a shortage of food in the neighborhood. The situation will be automatically resolved in 180 days.

FUELSHRT (Fuel shortage): There is a shortage of fuel (i.e., gasoline) in the neighborhood. The situation will be automatically resolved in 30 days.

GARBAGE (Garbage in the streets): Garbage is accumulating in the streets, causing health and safety issues. The situation will spawn a **DISEASE** situation in 2 days, and will be automatically resolved in 45 days.

INDSPILL (Industrial spill): There has been some kind of industrial spill in the neighborhood. The situation will spawn a **DISEASE** situation in 5 days, and will be automatically resolved in 90 days.

MINEFIELD (Minefield): There is a mine field in the neighborhood. The situation will be automatically resolved in 1080 days.

NOWATER (No water): There is no drinking water available in the neighborhood. The situation will spawn a **DISEASE** situation in 2 days, and will be automatically resolved in 3 days.

ORDNANCE (Unexploded ordnance): Unexploded ordnance (e.g., bombs) has been found in the neighborhood. The situation will be automatically resolved in 540 days.

PIPELINE (Oil pipeline fire): An oil pipeline is on fire in the neighborhood. The situation will be automatically resolved in 7 days.

POWEROUT (Power outage): There is a power outage in the neighborhood. The situation will be automatically resolved in 60 days.

REFINERY (Oil refinery fire): An oil refinery is on fire in the neighborhood. The situation will be automatically resolved in 5 days.

RELSITE (Damage to religious site): An important religious site has been damaged in the neighborhood, leading to unrest. The situation will be automatically resolved in 45 days.

SEWAGE (Sewage spill): Sewage is pooling in neighborhood streets, reducing quality of life and causing health issues. The situation will spawn a **DISEASE** situation immediately, and will be automatically resolved in 30 days.

14.8.2 Operations on Environmental Situations

There are two significant operations on environment situations:

- They can be created.
- They can be resolved.

Creating an environmental situation implies that a situation of the given type has just begun. This is generally a bad thing, and will cause an immediate decline in satisfaction at the next

time advance. This is called the *inception penalty*. Satisfaction will continue to decline so long as the situation continues, up until the situation is *resolved*, i.e., the problem is fixed. At this point the on-going satisfaction effects will cease, and there will a *resolution benefit* which makes satisfaction levels go up. Whether or not the resolution benefit offsets the previous decline depends on how soon resolution occurs.

In JNEM, these situations were only resolved manually, when the exercise controllers deemed that the training audience had put forth sufficient effort.

In Athena, environmental situations may be created explicitly by the analyst, or when an actor's **EXECUTIVE** tactic sends the **ENSIT:CREATE** order. Resolution may be scheduled when the situation is created, or may occur when an actor's **EXECUTIVE** tactic sends the **ENSIT:RESOLVE** order.

14.8.3 Attributes of Environmental Situations

Environment situations have the following attributes, which must be set when the situation is created:

| Attribute | Description |
|------------------------------------|--|
| Type (<i>stype</i>) | The type of environmental situation, as listed in Section 14.8.1. |
| Coverage (<i>coverage</i>) | The fraction of the neighborhood affected by the situation, as a real number from 0.0 to 1.0. |
| Inception (<i>inception</i>) | A Boolean flag; true if there should be an inception penalty when the situation is created, and false otherwise. Use false for situations that are on-going at time 0. |
| Caused By (<i>g</i>) | The name of the group that the neighborhood population considers responsible for this situation, or NONE . This information could be used to drive relationship or cooperation changes; at present, it is unused. |
| Resolved By (<i>resolver</i>) | The name of the group that will automatically resolve the situation, if automatic resolution is enabled. This information is used only to determine whether the resolver was a local or foreign group, which affects the Autonomy (AUT) effects in the rule sets. It could also be used to drive relationship or cooperation changes. The resolver can also be specified if the situation is resolved explicitly. |
| Duration (<i>rduration</i>) | The duration of the situation in days. If positive, the situation will be automatically resolved at the end of the interval, and the resolution will be attributed to the "Resolved By" group. If 0, no auto-resolution will take place. |

15. TACTIC TYPES

In Athena, actors use tactics to implement their decisions (Section 6.1.4). This section lists all of the available tactic types, with information about their parameters and how to make use of them.

15.1 ASSIGN

The **ASSIGN** tactic allows an actor to assign his own force or organization group personnel to perform a particular activity in a particular neighborhood. The personnel must already be deployed to the neighborhood, and must not have been assigned to do any other activity. In addition, the actor must have sufficient cash-on-hand to pay for it. The cost in dollars per person per week of assigning troops to do an activity is defined for each activity type by parameters in the Model Parameter database. Look under `activity.FRC.*.cost` and `activity.ORG.*.cost` for the cost parameters. By default, all activity costs are zero.

The assignment lasts for the following week, up until the next time strategies are executed. The assigned personnel will appear as units on Athena's **Map** tab.

If there are insufficient personnel, or insufficient funds, the tactic will not execute.

If the actor executes multiple **ASSIGN** tactics during the same week for the same neighborhood, group, and activity, the results are cumulative.

Activity assignment is described in Section 3.6; the effects of having the troops perform the activity are described in the *Athena Rules* document. See below for the list of possible activities.

The tactic parameters are as follows:

| Parameter | Description |
|---------------------|---|
| Group (g) | The ID of the force or organization group to be assigned an activity. |
| Neighborhood (n) | The ID of the neighborhood in which the activity will take place. |
| Activity (text1) | The name of the activity to be performed. Note that organization groups can perform only a few of the defined activities. |
| Personnel (int1) | The number of personnel to perform the activity. |

15.1.1 Activities

The activities that can be assigned to force and organization group personnel are as follows; see the *Athena Analyst's Guide* and the *Athena Rules Document* for more information.

| Activity | FRC | ORG | Description |
|------------------|-----|-----|--|
| CHECKPOINT | • | | Operating a checkpoint |
| CMO_CONSTRUCTION | • | • | Doing or facilitating construction |
| CMO_DEVELOPMENT | • | • | Encouraging light economic development |
| CMO_EDUCATION | • | • | Teaching local civilians |

| Activity | FRC | ORG | Description |
|---------------------|-----|-----|--|
| CMO_EMPLOYMENT | • | • | Employing local civilians |
| CMO_HEALTHCARE | • | • | Providing healthcare |
| CMO_INDUSTRY | • | • | Aiding local industry |
| CMO_INFRASTRUCTURE | • | • | Improving local infrastructure |
| CMO_LAW_ENFORCEMENT | • | | Enforcing the law/keeping the peace |
| CMO_OTHER | • | • | Other Civil/Military Operations, e.g., distributing food |
| COERCION | • | | Coercing cooperation through threats of violence |
| CRIMINAL_ACTIVITIES | • | | Engaging in criminal activities against enemies of the group |
| CURFEW | • | | Enforcing a curfew |
| GUARD | • | | Guarding sites |
| PATROL | • | | Patrolling the neighborhood |
| PSYOP | • | | Doing psychological operations |

15.2 ATTROE

The **ATTROE** tactic allows an actor to set the attacking ROE for his own force group in a particular neighborhood (Section 3.12); this will determine whether and who the group will choose to attack over the following week. By default, all groups have an ROE of **DO_NOT_ATTACK** in all neighborhoods.

Simply executing this tactic does not guarantee that any attacks will actually occur. Both the group and its enemy must be deployed to the neighborhood, and various other conditions must be made. In addition, the attacker must have sufficient cash-on-hand to pay for the requested number of attacks.

The tactic parameters are as follows:

| Parameter | Description |
|---------------------------------|---|
| Attacking Group (<i>f</i>) | The ID of the force group to be conducting attacks. |
| Defending Group (<i>g</i>) | The ID of the force group to be attacked. If the attacking group is uniformed, then this group must be non-uniformed, and <i>vice versa</i> . |
| Neighborhood (<i>n</i>) | The ID of the neighborhood in which the attacks may take place. |
| ROE (<i>text1</i>) | <p>The ROE proper. The set of possible values depends on whether the attacking group is uniformed or non-uniformed.</p> <p>Attacker is uniformed: The only choices are ATTACK and DO_NOT_ATTACK.</p> <p>Attacker is non-uniformed: The attacker has two modes of attack, HIT_AND_RUN and STAND_AND_FIGHT. If the ROE is HIT_AND_RUN, the attacker will try to nibble away at the defender while minimizing losses. Any potential attack that would result in too many losses will be foregone.</p> <p>If the ROE is STAND_AND_FIGHT, then the attacker will give up a certain number of casualties to kill as many enemies as possible.</p> |
| Max Attacks (<i>int1</i>) | The maximum number of attacks for which the actor is willing to pay over the next week. The cost of each attack is set for each force group during scenario preparation; see Section 14.4. The required amount of money is removed from the actor's cash-on-hand when the tactic executes; if not all attacks were carried out during the week, the unspent money is returned to the actor's cash-on-hand the next week. |

15.3 DEFROE

The **DEFROE** tactic allows the actor to set the defending ROE for one of his uniformed force groups in a particular neighborhood (Section 3.12). This determines how the group will defend itself when attacked in that neighborhood. Because this tactic simply sets the group's defensive response, it has no direct costs in either dollars or personnel. In the absence of a DEFROE tactic, every uniformed force group's defending ROE will be **FIRE_BACK_IF_PRESSED**; see below for a description of this ROE.

The primary effect of the defending ROE is on civilian collateral damage: the more aggressive the defense, the more civilians will be killed.

Only uniformed force groups have a defending ROE; it is assumed that uniformed force groups are better equipped than non-uniformed force groups, and that when a uniformed force group attacks a non-uniformed force group, it will do so with overwhelming force. In short, the non-uniformed force group's response is irrelevant.

The tactic parameters are as follows:

| Parameter | Description |
|------------------------|--|
| Defending Group (g) | The ID of the uniformed force group. |
| Neighborhood (n) | The ID of the neighborhood. |
| ROE (text1) | <p>The ROE proper. The set of values are as follows:</p> <p>HOLD_FIRE: Do not return fire. The attackers will escape, but no civilians will be killed.</p> <p>FIRE_BACK_IF_PRESSED: If the attackers HIT_AND_RUN, let them go. But if they STAND_AND_FIGHT, then return fire. The attackers will be killed, but so will some civilians.</p> <p>FIRE_BACK_IMMEDIATELY: Return fire immediately on any attack. The attackers will be killed, but more civilians will be killed.</p> |

15.4 DEMOB

The **DEMOB** tactic allows the actor to explicitly remove his force or organization group personnel from the playbox. For example, **DEMOB** could be used to model a withdrawal of U.S. troops from a particular region, when the appropriate withdrawal conditions were met. Only personnel who have not yet been deployed for the week can be demobilized.

Note that any troops remaining undeployed at the end of strategy execution will be demobilized automatically.

The tactic parameters are as follows:

| Parameter | Description |
|------------------------------|---|
| Group (<i>g</i>) | The ID of a force or organization group belonging to the actor. |
| Mode (<i>text1</i>) | The demobilization mode. If ALL , all of group <i>g</i> 's personnel will be demobilized. If SOME , then the number of personnel is determined by the "Personnel" parameter. |
| Personnel (<i>int1</i>) | The maximum number of personnel to demobilize when the "Mode" is SOME . If the group has fewer than this number of personnel remaining, all will be demobilized. |
| Once? (<i>once</i>) | A Boolean flag; whether to execute this tactic once only. If YES , this tactic will be considered for execution each week until it finally executes; then it will be disabled. ¹³ Suppose, for example, the actor wishes to withdraw half of his troops from the playbox as soon as some state of affairs is reached, but also wishes to leave the rest of his troops there indefinitely. He will not want to execute this tactic again the following week, so he would set this parameter to YES . If NO , the tactic will be able to execute repeatedly, so long as any attached conditions are met. |

¹³ The analyst can re-enable the tactic through the Strategy Browser if desired.

15.5 DEPLOY

The **DEPLOY** tactic allows the actor to deploy his force and organization group personnel into particular neighborhoods in the playbox. The personnel must already be present in the playbox, either remaining from the previous week or newly mobilized using the **MOBILIZE** tactic.

Available Troops: Athena's basic assumption is that the actor can deploy his troops wherever he likes from week-to-week, and that he makes his deployment decisions week-by-week. Before each strategy execution, consequently, it is as though all troops are withdrawn from all neighborhoods, and made available for deployment. It is these troops, along with any newly mobilized troops, that must be deployed each week.

Deployment Cost: Each force and organization group definition includes a deployment cost in dollars per person per week. This cost is incurred by the **DEPLOY** tactic; if the actor has insufficient funds, the troops will not be deployed (and will likely be demobilized).

Multiple Deployments: If multiple **DEPLOY** tactics execute for the same group and neighborhood, the results are cumulative.

Undeployed Troops: Troops remaining undeployed at the end of strategy execution are automatically demobilized, as though the **DEMOB** tactic was used. Thus, it is usually wise to have a catch-all **DEPLOY** tactic with mode **ALL**.

The tactic parameters are as follows:

| Parameter | Description |
|--------------------------------------|--|
| Group (<i>g</i>) | The ID of a force or organization group belonging to the actor. |
| Mode (<i>text1</i>) | The deployment mode. If ALL , all of group <i>g</i> 's that are still available for deployment will be deployed. If SOME , then the number of personnel is determined by the "Personnel" parameter. |
| Personnel (<i>int1</i>) | The number of personnel to deploy when the "Mode" is SOME . If the group has fewer than this number of personnel remaining, the tactic will not execute. (There is likely no point in deploying 500 personnel to a neighborhood when you really wanted 5000.) |
| In Neighborhoods (<i>nlist</i>) | The list of neighborhoods into which the personnel are to be deployed. The personnel will be distributed evenly across the neighborhoods. |

15.6 DISPLACE

The **DISPLACE** tactic allows the **SYSTEM** agent to displace the civilian population of a neighborhood from their homes. Like force and organization group assignments, the displacement lasts for the following week; for the displacement to be permanent, this tactic must execute each week. The displaced personnel will appear as units on Athena's **Map** tab.

Under the covers, displacement is simply a group activity, and is assessed by the same logic; see Sections 3.6, 3.9, and 3.10.

Subsistence Agriculture: Civilians engaged in subsistence agriculture can no longer do so once displaced; they are forced to enter the local economy as consumers, and possibly as workers as well.

The tactic parameters are as follows:

| Parameter | Description |
|---------------------|--|
| Group (g) | The ID of a civilian group. |
| Neighborhood (n) | The ID of the neighborhood into which the civilians will be displaced; this can be the group's home neighborhood. |
| Activity (text1) | The name of the activity to be performed, which determines just how the civilians will live. If DISPLACED , they will mingle with the local population: they will affect local attitudes, and may enter the work force. If IN_CAMP , they are put into displaced persons camps: they have no effect on local attitudes, being separated from the local population, and do not enter the work force. |
| Personnel (int1) | The number of personnel to be displaced. |

15.7 EXECUTIVE

The **EXECUTIVE** tactic allows any actor, or the **SYSTEM** agent, to execute an executive command during scenario execution. (See Section 10.8 information about executive commands.) **EXECUTIVE** tactics can write messages to the log, pause the simulation, and send many different orders via the **send** command. It is also possible to group executive commands into a script, and then call the script using the **call** command. For example, a script might model some kind of catastrophe by creating a magic attitude driver, and then creating satisfaction or cooperation inputs against it, essentially writing a DAM rule on the fly. See the cookbook in Part III for examples.

Note: Using an **EXECUTIVE** tactic is really a kind of programming; be sure to test your executive commands before you assume that they are working. Any errors will appear in the Significant Events Log when the simulation pauses; see `my : //app/sigevents` in the Detail Browser.

The tactic has the following parameters:

| Parameter | Description |
|--------------------|---|
| Command (text1) | The executive command to execute. |
| Once? (once) | <p>A Boolean flag; whether to execute this tactic once only. If YES, this tactic will be considered for execution each week until the attached conditions are finally met, and executes; then it will be disabled.¹⁴ Suppose, for example, the actor wishes to cause a catastrophe once, when conditions are first met. He will not want to execute this tactic again the following week, so he would set this parameter to YES.</p> <p>If NO, the tactic will be able to execute repeatedly, so long as any attached conditions are met.</p> |

¹⁴ The analyst can re-enable the tactic through the Strategy Browser if desired.

15.8 FUND

The **FUND** tactic allows an actor to give money to other actors. The funding actor may specify any amount of money, which will come out of the actor's cash-on-hand. It will be added to the recipient's cash-on-hand at the beginning of the subsequent strategy execution.¹⁵ Multiple **FUND** tactics with the same recipient during the same week are cumulative.

Each execution of the **FUND** tactic is a one-time event; but the tactic can be re-executed week after week to set up a continuing stream of money.

The tactic will not execute if the funding actor has insufficient funds in his cash-on-hand.

The tactic has the following parameters:

| Parameter | Description |
|-------------------------|---|
| Actor (a) | The ID of the actor to receive the funds. |
| Amount, \$/week (x1) | The amount of money, in dollars, to provide to the recipient. |

¹⁵ The actors are presumed to execute their strategies in parallel; but the time the money is available, the recipient has already executed their strategy for the week. Hence, the money becomes available the following week.

15.9 FUNDENI

The **FUNDENI** tactic allows an actor to fund Essential Non-Infrastructure (ENI) services for particular groups (Section 3.13). The actor may specify any amount of money, which will come out of the actor's cash-on-hand. The tactic will not execute if the funding actor has insufficient funds.

Effect on the Civilians: Civilian groups require a minimum amount of ENI services, and may expect a much higher amount. When a group's requirements and expectations are not met, the group's mood will fall, and the group's vertical relationships with the actors may also fall. If the group's requirements and expectations are exceeded, the group's mood and vertical relationships may rise.

Multiple Tactics: If multiple **FUNDENI** tactics are executed for a group, by the same or multiple actors, the effects are cumulative.

Saturation: Any given group can use only so much in the way of ENI services; money spent in excess of this saturation level is wasted. When multiple actors are funding ENI services for a group, and saturation is reached, the actor in control in the neighborhood is the first to get credit.

Continuous Funding: Each execution of the **FUNDENI** tactic provides funding for the following week. To provide continuous funding over time, the tactic must be executed each week.

The tactic has the following parameters:

| Parameter | Description |
|-------------------------|---|
| Groups (glist) | The IDs of the civilian groups to receive the funds. |
| Amount, \$/week (x1) | The amount of money, in dollars, to provide to the groups for ENI services. The groups receive the funds in proportion to their population; or, to put it another way, every individual belonging to one of the listed groups will receive the same dollar amount of service. |

15.10 MOBILIZE

Each force and organization group has an initial quantity of personnel in the playbox, based on the status quo deployment defined as part of the scenario inputs (Section 12.12); the **MOBILIZE** tactic allows the actor who owns a force or organization group to bring additional personnel into the playbox, e.g., to implement a surge or to reflect recruitment from the local population.¹⁶ Once mobilized, the troops can be **DEPLOY**ed as usual.

Tactic Priorities: Only **DEPLOY** tactics with lower priority than this tactic can make use of the newly mobilized personnel. Consequently, this tactic should usually have a higher priority than any **DEPLOY** tactic in the actor's strategy. Tactic priority can be adjusted in the Strategy Browser.

The tactic parameters are as follows:

| Parameter | Description |
|---------------------|--|
| Group (g) | The ID of a force or organization group belonging to the actor. |
| Personnel (intl) | The number of personnel to mobilize. |
| Once? (once) | A Boolean flag; whether to execute this tactic once only. If YES , this tactic will be considered for execution each week until it finally executes; then it will be disabled. ¹⁷ (You don't have a surge every week.) If NO , the tactic will be able to execute repeatedly, so long as any attached conditions are met; this might be appropriate to reflect recruiting from the local population. |

¹⁶ In the future, we expect to have explicit models of recruitment and desertion.

¹⁷ The analyst can re-enable the tactic through the Strategy Browser if desired.

15.11 SAVE

Each actor has two pots of money: his cash-on-hand and his cash reserve. Money in the first is available to be spent on tactics; money in the second is explicitly being saved for later. By managing his cash reserve using the **SAVE** and **SPEND** tactics, the actor can build up a war chest for a major offensive.¹⁸

The **SAVE** tactic transfers a percentage of the actor's income for the week from the actor's cash-on-hand into his cash-reserve. The tactic does not execute given insufficient funds; thus, this tactic should usually be prioritized higher than other tactics that consume money.

The tactic parameters are as follows:

| Parameter | Description |
|--------------------------------|---|
| Percentage of Income (int1) | The percentage of the actor's income, from 0 to 100, to transfer to his cash reserve this week. |

¹⁸ At least, that was the idea. Whether this feature will prove useful in practice is yet to be seen.

15.12 SPEND

Each actor has two pots of money: his cash-on-hand and his cash reserve. Money in the first is available to be spent on tactics; money in the second is explicitly being saved for later. By managing his cash reserve using the **SAVE** and **SPEND** tactics, the actor can build up a war chest for a major offensive.¹⁹

The **SPEND** tactic transfers a percentage of the actor cash reserve into his cash-on-hand, so that it can be spent on tactics. The tactic always succeeds, but if there's no money in the cash reserve no money is transferred.

The tactic parameters are as follows:

| Parameter | Description |
|--|---|
| Percentage of Reserve (<i>int1</i>) | The percentage of the actor's cash reserve, from 0 to 100, to release for spending this week. |

¹⁹ At least, that was the idea. Whether this feature will prove useful in practice is yet to be seen.

15.13 SUPPORT

The **SUPPORT** tactic allows the actor to give his political support to various actors on a neighborhood by neighborhood basis (Section 6.2). The actor's default allocation of support is part of the actor's definition; he may, by default, support himself, some other actor, or no actor, in all neighborhoods. This tactic allows him to tailor his support neighborhood by neighborhood. For example, an actor may support himself in one neighborhood, a political ally in a nearby neighborhood, and no one at all in any other neighborhood. By so doing, he indicates that he does not intend to be a political player in these other neighborhoods.

Support is given only for the following week, at which time it must be explicitly renewed if the support is to continue.

The tactic parameters are as follows:

| Parameter | Description |
|-----------------------------|--|
| Supported Actor (a) | The ID of the actor that will receive the actor's political support, or SELF , or NONE . |
| In Neighborhoods (nlist) | The IDs of the neighborhoods in which the actor's political support is given. |

16. CONDITION TYPES

Conditions are Boolean predicates about the state of the simulation that are used to control whether and when the tactics in actor's strategies are executed (Section 6.1.2). This section lists all of the conditions, explains their precise meanings, and documents their parameters.

The emphasis is on how to make use of these conditions rather than on precise data requirements, which are covered in the on-line help. For example, the on-line help covers the syntax of time specification strings, which are used to specify particular dates for the **AFTER**, **AT**, **BEFORE**, and **DURING** conditions.

16.1 AFTER

This condition is met when the current simulation time is later than a particular date, and unmet otherwise. The condition parameters are as follows:

| Parameter | Description |
|-------------|--|
| Day (t1) | The date after which the condition is met, specified as a time specification string. |

16.2 AT

In principle, this condition is met when the current simulation time is exactly some particular date $t1$, and unmet otherwise. However, conditions are only evaluated when strategies are to be executed, which happens once every seven days. Consequently, this condition will be met at the first strategy execution on or following time $t1$.

The condition parameters are as follows:

| Parameter | Description |
|-----------------|---|
| Day ($t1$) | The date on which the condition is met, specified as a time specification string. |

16.3 BEFORE

This condition is met when the current simulation time is earlier than a particular date, and unmet otherwise. The condition parameters are as follows:

| Parameter | Description |
|-------------|---|
| Day (t1) | The date before which the condition is met, specified as a time specification string. |

16.4 CASH

This condition compares an actor's cash reserve with some specified amount. The condition is met if the comparison is true, and unmet otherwise. Using this condition, the actor can undertake tactics when he has saved enough money. (See also the **SAVE** and **SPEND** tactics.)

The condition can query *any* actor's cash reserve, but an actor knows more about his own cash reserve than he does about that of other actors (Section 6.1.6). Thus, when he queries his own cash reserve, he queries its value **at that point in strategy execution**, rather than its value before strategy execution began.

For example, the actor might choose to execute a tactic if his cash reserve is greater than or equal to \$50,000.

The condition has the following parameters:

| Parameter | Description |
|---------------------|--|
| Actor (a) | The name of the actor whose cash reserve is being queried. |
| Comparison (op1) | <p>The kind of comparison to do:</p> <ul style="list-style-type: none"> • EQ: equal to • GE: greater than or equal to • GT: greater than • LE: less than or equal to • LT: less than <p>Note: All values are rounded to the penny before comparison.</p> |
| Amount (x1) | The amount of cash to which the actor's cash reserve will be compared. |

16.5 CONTROL

The **CONTROL** condition is met when an actor controls each of one or more listed neighborhoods, and is unmet otherwise (Section 6.2).

The condition has the following parameters:

| Parameter | Description |
|--------------------------|---|
| Actor (a) | The name of the actor whose control is being queried. |
| Neighborhoods (nlist) | A list of one or more neighborhood IDs. |

16.6 DURING

The **DURING** condition is met when the current simulation time is within a particular interval, specified by a start date and an end date, and unmet otherwise.

The condition parameters are as follows:

| Parameter | Description |
|-------------------|---|
| Start Day (t1) | The date which begins the interval, specified as a time specification string. |
| End Day (t2) | The date which ends the interval, specified as a time specification string. |

16.7 EXPR

The **EXPR** condition is met when a particular Boolean expression is true. The expression has TCL expression syntax, and is evaluated by the Athena executive (Section 10.8). The executive provides a large number of functions that can be used in **EXPR** expressions; see “Executive Functions” in the on-line help for the complete list.

For example, the expression

```
security('N1','G1') < 10 && mood('G1') < -15.0
```

will be true when the security of group G1 is less than 10 in neighborhood N1, and the mood of G1 is less than -15.0.

Note: Using **EXPR** expressions is really a kind of programming; be sure to test your expressions before you assume that they are working. Any errors will appear in the Significant Events Log when the simulation pauses; see `my://app/sigevents` in the Detail Browser.

The condition parameters are as follows:

| Parameter | Description |
|-----------------------|---|
| Expression (text1) | A Boolean expression with TCL syntax to be evaluated by the Athena executive. |

16.8 INFLUENCE

The **INFLUENCE** condition compares an actor's influence in a neighborhood with some specified amount. The condition is met if the comparison is true, and unmet otherwise. Using this condition, the actor can undertake tactics when he has enough influence to make it worthwhile, or to shore up influence that's declining (Section 6.2).

The condition has the following parameters:

| Parameter | Description |
|---------------------|---|
| Actor (a) | The name of the actor whose influence is being queried. |
| Neighborhood (n) | The neighborhood in which the actor might or might not have influence. |
| Comparison (op1) | <p>The kind of comparison to do:</p> <ul style="list-style-type: none"> • EQ: equal to • GE: greater than or equal to • GT: greater than • LE: less than or equal to • LT: less than <p>Note: All values are rounded to two places after the decimal before comparison.</p> |
| Amount (x1) | The amount of influence to which the actor's influence will be compared. |

16.9 MET

The **MET** condition is met when some specified set of the actor's goals are all met, and is unmet otherwise. Attach this condition to a tactic to take action to maintain the state of affairs represented by the goals that are met.

Note that this condition can only be attached to tactics, not to goals.

The condition has the following parameters:

| Parameter | Description |
|------------------|--|
| Goals (list1) | The IDs of one or more of the actor's own goals. |

16.10 MOOD

The **MOOD** condition compares a civilian group's mood with a particular value. The condition is met if the comparison is true, and unmet otherwise. Using this condition, the actor can undertake tactics to improve the group's mood when the group is dissatisfied, or to take advantage of the group's mood when the group is satisfied (Section 5.4).

The condition has the following parameters:

| Parameter | Description |
|---------------------|---|
| Group (g) | The ID of the group whose mood is being queried. |
| Comparison (op1) | <p>The kind of comparison to do:</p> <ul style="list-style-type: none">• EQ: equal to• GE: greater than or equal to• GT: greater than• LE: less than or equal to• LT: less than <p>Note: All values are rounded to two places after the decimal before comparison.</p> |
| Amount (x1) | The mood to which the group's mood will be compared. |

16.11 NBCOOP

The **NBCOOP** condition compares a neighborhood's cooperation with a force group to some particular value. (The cooperation of a neighborhood with a force group is the average of the cooperation of the civilian groups in the neighborhood with that force group, weighted by their populations.) The condition is met if the comparison is true and unmet otherwise. Using this condition, the actor can undertake tactics to improve the neighborhood's cooperation when it is low, or to take advantage of it when it is high (Section 5.5).

The condition has the following parameters:

| Parameter | Description |
|------------------------------|--|
| Neighborhood (<i>n</i>) | The ID of the neighborhood whose cooperation is being queried. |
| Group (<i>g</i>) | The ID of the force group whose cooperation is being queried |
| Comparison (<i>op1</i>) | <p>The kind of comparison to do:</p> <ul style="list-style-type: none"> • EQ: equal to • GE: greater than or equal to • GT: greater than • LE: less than or equal to • LT: less than <p>Note: All values are rounded to one place after the decimal before comparison.</p> |
| Amount (<i>x1</i>) | The cooperation to which the neighborhood's cooperation will be compared. |

16.12 NBMOOD

The **NBMOOD** condition compares a neighborhood's mood with a particular value. (The mood of a neighborhood is the average of the moods of the civilian groups in the neighborhood, weighted by their populations.) The condition is met if the comparison is true and unmet otherwise. Using this condition, the actor can undertake tactics to improve the neighborhood's mood when its residents are dissatisfied or to take advantage of the neighborhood's mood when its residents are satisfied (Section 5.4).

The condition has the following parameters:

| Parameter | Description |
|---------------------|---|
| Neighborhood (n) | The ID of the neighborhood whose mood is being queried. |
| Comparison (op1) | <p>The kind of comparison to do:</p> <ul style="list-style-type: none"> • EQ: equal to • GE: greater than or equal to • GT: greater than • LE: less than or equal to • LT: less than <p>Note: All values are rounded to two places after the decimal before comparison.</p> |
| Amount (x1) | The mood to which the neighborhood's mood will be compared. |

16.13 TROOPS

The **TROOPS** condition compares a force or organization group's total personnel in the playbox with some particular value. For example, an actor might step up recruitment for a group if its total personnel drops below some threshold.

The behavior of the condition changes if the group belongs to the actor whose strategy is being executed. If it does not, the condition is based on the group's total personnel prior to the start of strategy execution. If it does, the condition takes into account any mobilization or demobilization decisions already made by the actor at this point in strategy execution.

The condition has the following parameters:

| Parameter | Description |
|------------------------------|--|
| Group (<i>g</i>) | The ID of the force or organization group. |
| Comparison (<i>op1</i>) | The kind of comparison to do: <ul style="list-style-type: none"> • EQ: equal to • GE: greater than or equal to • GT: greater than • LE: less than or equal to • LT: less than |
| Amount (<i>int1</i>) | The number to which the group's personnel will be compared. |

16.14 UNMET

The **UNMET** condition is met when at least one of a specified set of the actor's goals is unmet, and is unmet otherwise. Attach this condition to a tactic to take action towards one or more goals just so long as the goals are unmet.

Note that this condition can only be attached to tactics, not to goals.

The condition has the following parameters:

| Parameter | Description |
|------------------|--|
| Goals (list1) | The IDs of one or more of the actor's own goals. |

17. SIMULATION ORDERS

Inputs to the Athena scenario and models are presented to Athena as *orders*.

Orders may be sent interactively by a variety of methods; such orders can usually be undone. The complete history of orders sent interactively for the current scenario is shown on the **Orders/Sent** tab.

In addition, orders can be scheduled in advance. Scheduled orders are executed as time advances, and consequently cannot be undone. The full set of orders that are waiting for execution is shown on the **Orders/Scheduled** tab.

Finally, orders can be sent and scheduled by the **EXECUTIVE** tactic.

In future releases, this section should contain complete documentation of all Athena orders. For now, the user is referred to the on-line help. Select **Help/Orders** from the menu, or "Orders" or its children from the "?" tab in the Detail Browser's sidebar.

18. GLOSSARY

This section contains a glossary of terms. When a glossary term is included in another term's definition, it is printed in *italics*.

activity

An *activity* is an action that a *group* can perform in a *neighborhood*, as initiated by a *tactic*. Activities are staffed by personnel deployed to neighborhoods during *strategy* execution. Personnel assigned to an activity appear as a *unit* on the **Map** tab; the unit is said to perform the activity. There are different sets of activities for the different group types: *force groups*, *organization groups*, and *civilian groups*. For the full list of activities, and the meaning and effect of each, see the *Athena Rules* document.

actor

A significant decision maker in the *playbox*. See Sections 3.3 and 14.2.

affinity

The affinity of one *group* or *actor* for another is a measure of the depth of feeling or support of the one for the other. Affinity is a number from +1.0 to -1.0; it is computed by comparing the *belief systems* of the two entities, and is the basis for group and group/actor *relationships*. Affinity is often described using the following scale:

| Name | Long Name | Value | Bounds |
|---------|-------------------|-------|--------------------------|
| SUPPORT | Supports | 0.8 | $0.7 < value \leq 1.0$ |
| LIKE | Likes | 0.4 | $0.2 < value \leq 0.7$ |
| INDIFF | Is Indifferent To | 0.0 | $-0.2 < value \leq 0.2$ |
| DISLIKE | Dislikes | -0.4 | $-0.7 < value \leq -0.2$ |
| OPPOSE | Opposes | -0.8 | $-1.0 < value \leq -0.7$ |

attitude

The *attitude* of a particular *group* toward something. Attitudes include group *satisfaction* with respect to the four *concerns*, the *cooperation* of *civilian groups* with *force groups*, the *horizontal relationships* between all pairs of groups, and the *vertical relationships* between civilian groups and *actors*.

attitude driver

1. Any *event* or *situation* occurring in the *playbox* that causes a change in the *attitudes* of members of one or more *civilian groups*. Examples of drivers include casualties to civilian groups, the presence and activities of force group in neighborhoods, environmental situations, demographic situations, and magic attitude drivers.

2. A *magic attitude driver*.

3. A *GRAM* driver ID, used by *GRAM* to track individual drivers. This ID is useful when examining the *GRAM* data in the *RDB*.

Autonomy (AUT)

One of the four *concerns*. See Section 5.4.1.

belief system

Each *civilian group* and each *actor* has a belief system, which is a set of beliefs with regard to a number of *topics*. Each belief consists of a *position* for or against the topic, along with an *emphasis* on agreement or disagreement for that position.

Belief systems are used to compute *affinities*, which in turn are the basis for group and group/actor relationships.

bgerror

When Athena encounters an unexpected run-time error, it does not usually halt execution. Instead, the error and a complete stack trace are written to the *debugging log*, and the user is notified. This is called a *background error*, or *bgerror* for short. See Section 10.7 for details.

calibration

Athena's models can be calibrated by tweaking Athena's *Model Parameters*. This is particularly important for the Economics model; see Section 13.2 for the steps to calibrate the economy.

capacity

Each *sector* in the Economics model has a maximum size, or *capacity*. The size of the **pop** sector is limited by the size of the labor force, for example.

Usually when we speak of capacity, however, we mean the capacity of the **goods** sector: the ability of the economy to produce goods and services. This capacity is set when the scenario is locked and can then be increased or decreased *neighborhood* by neighborhood by setting the Production Capacity Factor (PCF) on the **Econ/Capacity** tab.

cause

Every *attitude driver* has an associated *cause*. All *inputs* to *GRAM* for the driver will be tagged with this cause; for example, the **DISEASE** and **EPIDEMIC** *environmental situations* both have the cause **SICKNESS**.

The purpose of causes is to prevent multiple inputs of a similar type from piling on in an unreasonable way. For example, suppose that there is a serious epidemic in *neighborhood A*

which then spreads to nearby neighborhood B. While the epidemic remained in A, the folks in B were somewhat unhappy about it (because it might spread, and because they care about some or all of the people in A). Once the epidemic spreads to B, though, the people in B have their own problems; it's unlikely to expect them to feel all that bad about the situation in neighborhood A.

Next, consider neighborhood C, which is near to both A and B. The spread of the epidemic to B might heighten their alarm somewhat, but it shouldn't double their alarm.

In these cases, where there are multiple inputs to GRAM for a given *satisfaction* curve (say, group G's *SFT* in neighborhood C), the smaller inputs get swamped by the larger rather than adding to them.

The cause associated with each kind of driver can be found in the *Athena Rules* document.

civilian group

A *group* that represents civilian population in a *neighborhood*. See Sections 3.4.1 and 14.3.

command

A TCL *command* used to script Athena through the Athena *executive*. See Section 10.8.

concern

Athena tracks the *satisfaction* of *civilian groups* with respect to a number of *concerns*, also known as *soft factors*. See Section 5.4.

condition

A Boolean condition regarding the current *state* of the simulation which can be met or unmet. Conditions are used to define *goals*, and to control the execution of *tactics*. See Section 16.

consumer

A *consumer* is a member of the civilian population who consumes goods produced by the *local economy*. Consumers are thus distinguished from members of the *subsistence agriculture population*, who live off of their own land and do not participate in the local economy. Some consumers are also *workers*.

control

Actors with sufficient *support* in a *neighborhood* gain *influence* in that neighborhood; and given sufficient influence, may take *control* of the neighborhood. See Section 6.2.

cooperation

A cooperation level is an *attitude* that represents the likelihood that a member of a *civilian group* will give information to a member of a particular *force group*. Cooperation levels are represented using the following scale:

| Name | Long Name | Value | Bounds |
|------|------------------------|-------|---------------------------|
| AC | Always Cooperative | 100.0 | $99.9 < value \leq 100.0$ |
| VC | Very Cooperative | 90.0 | $80.0 < value \leq 99.9$ |
| C | Cooperative | 70.0 | $60.0 < value \leq 80.0$ |
| MC | Marginally Cooperative | 50.0 | $40.0 < value \leq 60.0$ |
| U | Uncooperative | 30.0 | $20.0 < value \leq 40.0$ |
| VU | Very Uncooperative | 10.0 | $1.0 < value \leq 20.0$ |
| NC | Never Cooperative | 0.0 | $0.0 < value \leq 1.0$ |

Athena creates the required cooperation curves automatically as groups are created. The cooperation between groups is viewed and modified on the **Attitudes/Cooperation** tab. Cooperation can also be changed by *attitude drivers* that trigger Athena's *DAM Rules*.

coverage

Coverage is a measure of the fraction of a *neighborhood* that is affected by something—a group *activity*, an *environmental situation*, etc.—expressed as a number between 0.0 and 1.0. It is a common input to the *DAM Rules*.

Culture (CUL)

One of the four *concerns*. See Section 5.4.1.

DAM Rules

Athena assesses the effect of simulation *drivers* using a collection of rule sets that belong to Athena's Driver Assessment Model (DAM); the rule sets are collectively known as the *DAM Rules*. The full set of *rules* and their effects are defined in the *Athena Rules* document.

day

Athena measures simulation time in integer days; each day represents one day of simulated calendar time.

Athena does not track any time interval shorter than one day; however, it is often convenient to speak of the sequence of events during a simulated day in terms of *morning*, *noon*, and *evening*:

Time can be thought to advance at midnight. After the time advance, in the "morning", Athena executes all scheduled *orders*. At "noon" the user has the opportunity to send orders, provided that Athena enters the **Paused state**. In the "evening", the current state of affairs is assessed: *security* and *coverage* are computed, *DAM Rules* fire, and so forth. Then, at midnight, time advances, and the cycle continues.

debugging log

As Athena runs, it writes a detailed log of its behavior, primarily intended for use in debugging the application and its models. This is distinct from the *Significant Events Log*, which logs significant simulation events. See Section 10.5.

demographic situation

A *demographic situation*, or *demsit*, is an on-going situation involving the residents of a neighborhood as detected by the Demographics model. It is an *attitude driver*. At present, there is only one type of demographic situation, the Unemployment (UNEMP) situation; see the *Athena Rules* document for details.

Existing demographic situations can be seen on the **Neighborhoods/DemSits** tab.

demsit

A *demographic situation*.

driver

An *attitude driver*.

effects delay

The delay, in *days*, before the effects of an *attitude driver* in *neighborhood N* are felt in neighborhood *M*; closely related to *proximity*. Effects delays are deprecated in Athena 3.

emphasis

In a *civilian group* or *actor's belief system*, a belief about a *topic* is represented as a *position* and an *emphasis* on agreement or disagreement with that position. The emphasis is a number from 0.0 to 1.0. When computing *affinity*, 0.0 indicates that disagreement counts entirely and agreement not at all, and 1.0 indicates that agreement counts entirely and disagreement not at all. In general, the lower the emphasis, the lower the resulting affinities.

Note: the extreme values of 0.0 and 1.0 represent extreme pathological cases and should not be used.

Emphasis should be defined using the following symbolic constants:

| Name | Long Name | Value |
|---------|----------------------|-------|
| ASTRONG | Agreement--Strong | 0.9 |
| AWEAK | Agreement | 0.7 |
| NEITHER | Neither | 0.5 |
| DWEAK | Disagreement | 0.35 |
| DSTRONG | Disagreement--Strong | 0.25 |

| | | |
|----------|-----------------------|------|
| DEXTREME | Disagreement--Extreme | 0.15 |
|----------|-----------------------|------|

ensit

An *environmental situation*.

environmental situation

An *environmental situation*, or *ensit*, is an on-going *situation* within a *neighborhood*; it is an *attitude driver*, and affects all of the people living in the neighborhood. It may also affect surrounding neighborhoods. For a complete list of situation types, and the effect of each, see the *Athena Rules* document.

evening

The end of a simulated *day*, when the current state of affairs is assessed.

event

1. An *attitude driver* that occurs at some particular point in time.
2. A significant event, as included in the *Significant Events Log*.
3. Any occurrence in the simulation.

executive

The Athena *executive* is the object that processes *commands* entered at the Athena command-line or in Athena *executive scripts*. In addition, it is responsible for executing **EXECUTIVE** *tactics* and for evaluating the **EXPR** *condition's* Boolean expressions.

executive script

A script of *commands* to be processed by the Athena *executive*, e.g., via an **EXECUTIVE** *tactic*.

force group

A *group* that exists in order to project force on behalf of an *actor*, e.g., the U.S. Army. See Sections 3.4.2 and 14.4.

goal

A state of affairs that an *actor* wishes to bring about (or to preserve if the state of affairs already exists). Goals are defined in terms of *conditions*. Actors can execute *tactics* to achieve or preserve *goals* via the **MET** and **UNMET** conditions. See Section 6.1.3.

GRAM

The Generalized Regional Attitude Model (GRAM) is that part of Athena that tracks *cooperation* and *satisfaction* levels, and the spread of *DAM Rule* inputs across the playbox.

group

A related collection of people in the *playbox*. There are three kinds of *group*: *civilian groups*, *force groups*, and *organization groups*. See Section 3.4.

horizontal relationship

The *relationship* of one *group* with another, expressed as a number from -1.0 to +1.0. Horizontal relationships need not be symmetric—that is, group F might like group G more or less than group G likes group F. Relationships are often described using the following scale:

| Name | Long Name | Value | Bounds |
|---------|-------------------|-------|--------------------------|
| SUPPORT | Supports | 0.8 | $0.7 < value \leq 1.0$ |
| LIKE | Likes | 0.4 | $0.2 < value \leq 0.7$ |
| INDIFF | Is Indifferent To | 0.0 | $-0.2 < value \leq 0.2$ |
| DISLIKE | Dislikes | -0.4 | $-0.7 < value \leq -0.2$ |
| OPPOSE | Opposes | -0.8 | $-1.0 < value \leq -0.7$ |

influence

Influence is a measure of an *actor's* influence in a *neighborhood*, based on the *support* for the actor by the residents of the neighborhood relative to other actors. Mathematically, influence is a fraction of neighborhood *control*; the influence of all actors in the neighborhood sums to one. In general, an actor takes control of a neighborhood by having an influence greater than 0.5.

input

1. Any value given to Athena as an input.
2. More specifically, a *satisfaction* or *cooperation* input given to *GRAM* by one of the *DAM Rules*.

JNEM

The Joint Non-kinetic Effects Model (JNEM) is a predecessor of JNEM used to add civilian responses to federations of battlefield simulations. Some of the models in Athena, notably *GRAM* and the *DAM Rules*, have their roots in JNEM.

labor force

The *labor force* are the *workers* in the *local economy*.

level input

An *input* to *GRAM* that causes a one-time change to a particular *satisfaction* or *cooperation* level and may have indirect effects that spread across the *playbox*.

local economy

Athena's Economics model concerns the economy of a region: a single country, part of that country, or perhaps several small countries taken together. Athena's *neighborhoods* can be "local", i.e., part of the economic region of interest, or non-local, outside the region.

The *local economy* is then the economy of this local region of interest.

lock

To *lock* the *scenario* is to leave the Scenario Preparation *state*, initialize the simulation, and make it possible for simulation time to advance. See also *unlock*.

log level

Every entry written to the *debugging log* has an associated severity, as follows:

- **fatal**: Used when the program is about to halt. Displayed with a red background.
- **error**: Used only for *bgerrors*. Displayed with an orange background.
- **warning**: Used when a potential problem is noticed. Displayed with a yellow background.
- **normal**: Normal informational message.
- **detail**: Detailed informational message.
- **debug**: Debugging message, low-level details. Note that **debug** messages are only written to the log when enabled by the developer.

The Debugging Log Browser's *log level* determines which messages are actually displayed. By default the log level is **normal**, meaning that the **Log** tab messages with severity **normal** and above.

magic attitude driver

A *magic attitude driver* (MAD) is a "magic" *driver* defined by the user; it affects *satisfaction* or *cooperation* levels in any way the user determines.

map

1. A visualization of the *playbox*, including *neighborhoods*, *group* personnel represented as *units*, and other data, as displayed on the **Map** tab.

2. An image file used as a background on the **Map** tab as an aid to visualization.

map reference string

A map reference string is a six-character string that represents a point on the *scenario's* map. See Section 11.6.2.

model parameter

An *input* to Athena that is used to calibrate or otherwise control Athena's models. *Model parameters* are stored in the model parameter database, which is part of the *scenario*.

Select **Help/Model Parameters** from the Athena menu to browse a description of the existing model parameters; select **Model Parameters** or one of its subheadings from the **Detail Browser**'s Objects tree to see default and current parameter values, and to change parameter values. Also, see Section 12.16.

mood

1. A *civilian group's satisfaction* averaged across its *concerns*, and taking *saliency* into account. In other words, how satisfied or dissatisfied the group is with things in general.
2. A *neighborhood's mood*, i.e., the average mood across all civilian groups in the neighborhood.

morning

The beginning of a simulated *day*, when scheduled *orders* are executed.

neighborhood

A region of the *playbox* in which *civilian groups* live and other *groups* conduct *activities*. See Section 3.2.

noon

The middle of a simulated *day*, when (if the simulation is **Paused**) the user has the opportunity to enter *orders*.

order

Inputs to the Athena *scenario* and models are presented to Athena as *orders*.

Orders may be sent interactively by a variety of methods; such orders can usually be undone. The complete history of orders sent interactively for the current scenario is shown on the **Orders/Sent** tab.

In addition, orders can be scheduled in advance. Scheduled orders are executed as time advances, and consequently cannot be undone. The full set of orders that are waiting for execution is shown on the **Orders/Scheduled** tab.

Finally, orders can be sent by the **EXECUTIVE** *tactic*.

organization group

A *group* representing an organization that is active in the *playbox*, e.g., for humanitarian relief. See Section 3.4.3.

parameter

1. A *model parameter*.
2. An input to a *condition*, *order*, or *tactic*.

playbox

The *playbox* is the region of interest to the analyst in the current *scenario*, i.e., the region for which *neighborhoods* are defined.

position

In a *civilian group* or *actor's belief system*, the entity's belief about a *topic* is represented as a *position* and an *emphasis* on agreement or disagreement with that position. The position is a number from 1.0 to -1.0 that represents not only whether the entity is for or against the topic, but also the strength of that position: the greater the absolute magnitude of the number, the more likely the entity is to take political action in support of its belief. Note that political action does not necessarily imply violent action.

Note: the extreme values of -1.0 and 1.0 represent extreme pathological cases and should not be used.

Positions can be defined using the following symbolic constants:

| Name | Long Name | Value | Bounds |
|------|----------------------|-------|----------------------------|
| P+ | Passionately For | 0.9 | $0.8 < value \leq 1.0$ |
| S+ | Strongly For | 0.6 | $0.45 < value \leq 0.8$ |
| W+ | Weakly For | 0.3 | $0.05 < value \leq 0.45$ |
| A | Ambivalent | 0.0 | $-0.05 < value \leq 0.05$ |
| W- | Weakly Against | -0.3 | $-0.45 < value \leq -0.05$ |
| S- | Strongly Against | -0.6 | $-0.8 < value \leq -0.45$ |
| P- | Passionately Against | -0.9 | $-1.0 < value \leq -0.8$ |

proximity

The *proximity* of one *neighborhood* with another is expressed as one of the following four values, **HERE**, **NEAR**, **FAR**, and **REMOTE**, where **HERE** represents the proximity of a neighborhood to itself.

The attitudes of a *civilian group* can be affected by *attitude drivers* that occur **HERE** in the same neighborhood; to a lesser degree by drivers occurring in a **NEAR** neighborhood, even less by drivers in a **FAR** neighborhood, and not at all by drivers in a **REMOTE** neighborhood.

Proximity is assigned by the analyst, and can take into account physical distance, physical obstacles, psychological distance, and so forth. Further, it need not be symmetric: neighborhood A might regard neighborhood B as **NEAR** while B regards A as **FAR**, or even as **REMOTE**.

Quality of Life (QOL)

One of the four *concerns*. See Section 5.4.1.

relationship

The *attitude* a *group* has toward another group (the *horizontal relationship*) or toward an actor (the *vertical relationship*).

ROE

In normal military usage, a unit's Rules of Engagement (*ROEs*) are the rules that determine when they may engage the enemy. In Athena, ROEs apply to a *force group* in a *neighborhood* and determine when and how the group can engage other groups. There will be no armed conflict, and hence no attrition or civilian casualties, unless the user creates the appropriate ROEs. ROEs are established using the **ATTROE** and **DEFROE** *tactics*.

rule

A small model that assesses the *state* of the simulation and gives SME-based input to another model. The term is especially used of the *DAM Rules*; however, many other models also contain rules.

rule set

A collection of one or more *rules*. In some cases, as in the *DAM Rules*, there really are individual rules in a rule set; in other cases, a rule set consists of a look-up table which is assumed to virtually represent a family of individual rules.

Run-time Database (RDB)

Most of Athena's working data is stored in the *Run-time Database* (RDB). It can be queried directly while Athena is running by using the **rdb** and **select** *commands*.

Safety (SFT)

One of the four *concerns*. See Section 5.4.1.

saliency

The *saliency* of a *concern* to a *group* is the importance of that concern to the group. Some groups will value *Quality of Life* more than *Autonomy*; for others, *Culture* will trump *Safety*. This comes into play when computing a group or *neighborhood's mood*, and when comparing *satisfaction* levels across groups.

Saliency is represented using the following scale:

| Name | Long Name | Value |
|------|----------------|-------|
| CR | Crucial | 1.000 |
| VI | Very Important | 0.850 |

| | | |
|----|----------------|-------|
| I | Important | 0.700 |
| LI | Less Important | 0.550 |
| UN | Unimportant | 0.400 |
| NG | Negligible | 0.000 |

satisfaction

A *satisfaction* level is an *attitude* that represents a *group's* satisfaction with respect to some *concern*. Strong positive or negative satisfaction will encourage a group to take action. Satisfaction levels are represented using the following scale:

| Name | Long Name | Value |
|------|-------------------|-------|
| VS | Very Satisfied | 80.0 |
| S | Satisfied | 40.0 |
| A | Ambivalent | 0.0 |
| D | Dissatisfied | -40.0 |
| VD | Very Dissatisfied | -80.0 |

scenario

An Athena *scenario* is the complete collection input data required to run the simulation.

scenario file

Athena is a document-centric application, rather like a word processor or spreadsheet application. An Athena document is called a *scenario file*. The scenario file contains the *scenario* data, the *map* (if any), reports, *snapshots*, and in general everything to do with the scenario. Athena scenario files have an ".adb" file type.

sector

Athena's Economics model divides the economy into three sectors, as described in Section 7.2. The three sectors are **goods**, **pop**, and **else**.

security

A *group's security* is computed by the Security model. It is a number that indicates how safe members of the group are in a *neighborhood*, as opposed to how satisfied they are with their safety. Security is represented by the following scale:

| Name | Long Name | Value | Bounds |
|------|-----------|-------|-------------------------|
| H | High | 60 | $25 < value \leq 100$ |
| M | Medium | 15 | $5 < value \leq 25$ |
| L | Low | -10 | $-25 < value \leq 5$ |
| N | None | -60 | $-100 < value \leq -25$ |

A group's security is affected by the *volatility* of the neighborhood.

service situation

An *attitude driver* that involves the provision of some service to the civilian population. At present, the only service situation is the Essential Non-Infrastructure (ENI) services situation; see Section 3.13.

Significant Events Log

As the Athena simulation runs, it writes significant simulation *events* to the Significant Events Log. This log can be browsed in the **Detail Browser**; see Sections 10.5 and 11.5. The Significant Events Log is distinct from the *debugging log*.

situation

An *attitude driver* that is on-going for two or more days. More particularly, an *activity situation*, a *demographic situation*, an *environmental situation*, or a *service situation*.

slope input

An *input* to GRAM that causes an on-going daily change to a particular *satisfaction* or *cooperation* level and may have indirect effects that spread across the *playbox*.

snapshot

Each time the user presses the *Run Simulation* button, the simulation saves a *snapshot* of its current *state*. When the simulation is paused, the user can use the VCR buttons on the main toolbar to return to any previous snapshot. This is called being in "snapshot mode". While in snapshot mode, the user can browse the various application tabs freely, but cannot make changes. See Section 11.3.2.

soft factors

The four *concerns*.

SQLite

An SQL database engine that stores its data in a single disk file. Athena uses SQLite to store its *run-time database* and its *scenario files*. SQLite is easily available and widely used in industry; users may find it convenient to use it to extract Athena outputs from scenario files.

state

1. The complete state of the simulation at a given time: all data associated with the simulation. This is what is saved in the *scenario file*.
2. The simulation mode in which Athena is currently operating. The Athena simulation can be in one of the following states at any given time:
 - Prep

- Running
- Paused
- Snapshot

See Section 10.2.

strategy

An *actor's strategy* is the collection of *tactics* he will use to achieve his *goals*, along with the set of *conditions* under which he will use them. The actor's strategy is browsed and edited on the **Strategy** tab.

subsistence agriculture population

That part of the population that lives by subsistence agriculture, whether farming or herding, and hence does not participate in the cash economy. Athena assumes that there is a hard line between the subsistence farmers and the *consumers*, although there is certainly some overlap in reality.

support

An *actor's support* in a *neighborhood* is the degree to which the people in the neighborhood (including *force group* and *organization group* personnel) will do his bidding. Support is based on the *vertical relationship* between the groups and the actor, on each group's *security* in the neighborhood, and on the number of people in each group. Support is used to compute *influence*.

tactic

An action that can be taken by an *actor* as part of his *strategy* to achieve his *goals*. A tactic will usually have *conditions* attached to it that will determine when it can be executed. See Sections 6.1 and 15.

tick

Athena advances time *day* by day; the processing done for each day as the simulation runs is referred to as happening at or on the "tick".

Some processing, such as computing attrition, happens every so many ticks (typically seven days, or one week); this weekly interval is sometimes referred to as the *tock*, e.g., "attrition is assessed every attrition tock".

time specification string

A time specification string identifies a particular simulation time. It has the form "*baseTime* +/- *offset*", where the offset is some number of *days* and the base time has one of the following forms:

- **T0**: Time zero, the beginning of the simulation.
- **NOW**: The current simulation time.

- **days**: The time as some number of days since **T0**.
- **zulu**: A *zulu-time* string.

Either the base time or the offset may be omitted. If the base time is omitted, it defaults to **NOW**; if the offset is omitted, it is 0. For example,

- **NOW**: The current simulation time.
- **NOW+30**: 30 days from now.
- **200000ZFEB10+7**: 7 days after 20 February 2010.
- **-5**: Five days ago
- **+5**: Five days from now
- **5**: Day five.

tock

See *tick*.

topic

A *civilian group* or *actor's belief system* consists of beliefs about a number of *topics*. A topic is something that unites or divides people, and which can drive people in the playbox to political action: e.g., Shia Islam or the presence of the US forces in the playbox.

There are an unlimited number of topics in any given region of the world. Topics are of interest from Athena's point of view if feelings for or against the topic are strong enough to motivate political action, and if there is strong disagreement about the topic among the groups and actors in the playbox.

Topics must be stated in absolute terms, so that they mean the same thing to all actors and groups. For example, "I control the country" is relative to the actor or group, and so is not a topic. "Actor A controls the country", on the other hand, is an absolute statement and so is a topic.

unit

A *unit* is a collection of personnel belonging to some *group* that can be positioned in the *playbox* and displayed on the **Map** tab.

Units come into existence automatically to represent the *neighborhood* population; to represent *force group* and *organization group* personnel deployed to neighborhoods by the **DEPLOY** tactic; to represent personnel assigned to particular *activities* by the **ASSIGN** tactic; and to represent civilians displaced from their home by the **DISPLACE** tactic. Thus, every unit belongs to some group.

Each unit is tied to a particular neighborhood and can be moved about on the map within that neighborhood for the purposes of visualization; however, a unit's specific location

within a neighborhood has no effect on the simulation. Units cannot be moved from one neighborhood to another.

In conventional war games, units correspond to a Table of Organization and Equipment (TOE); thus, a unit might be Company A of such-and-such a battalion. In Athena, a unit is simply an allocation of personnel to some purpose over a period of time, e.g., an allocation of members of a force group to the activity of law enforcement.

unlock

To *unlock* the scenario is to terminate a simulation run and re-enter the Scenario Preparation *state*, thus making it possible to edit the scenario inputs. Unlocking the scenario erases all of the current simulation results; if they should be retained, then save a copy of the *scenario file* prior to unlocking the scenario.

urbanization

The urbanization level of the neighborhood: **Isolated**, **Rural**, **Suburban**, or **Urban**. See Section 14.1.

vertical relationship

The *relationship* between a *civilian group* and an *actor*, expressed as a number from -1.0 to +1.0. Vertical relationships change over time, based on circumstances. Relationships are often described using the following scale:

| Name | Long Name | Value | Bounds |
|---------|-------------------|-------|--------------------------|
| SUPPORT | Supports | 0.8 | $0.7 < value \leq 1.0$ |
| LIKE | Likes | 0.4 | $0.2 < value \leq 0.7$ |
| INDIFF | Is Indifferent To | 0.0 | $-0.2 < value \leq 0.2$ |
| DISLIKE | Dislikes | -0.4 | $-0.7 < value \leq -0.2$ |
| OPPOSE | Opposes | -0.8 | $-1.0 < value \leq -0.7$ |

volatility

The *volatility* of each *neighborhood* is a number from 0 to 100 that indicates how likely it is that a random person in the neighborhood will get caught up in violence unrelated to his presence; as such, it is a major component of a *group's security* in the neighborhood.

worker

A member of the *labor force*, as determined by the Demographics model. All *workers* are *consumers*.

zulu-time

A zulu-time string, also known as a date-time group, represents a particular date and time in this format:

ddhhmmZmmmyy

where

dd The day of the month.

hhmm The time of day in hours and minutes, using a 24-hour clock, UTC (Greenwich Mean Time). (The "Z" indicates the time zone.) Because Athena measures time in days, this field is conventionally set to "0000" in all cases.

mmm The month of the year, specified as the first three letters of the month's name, e.g., "JAN", "FEB", etc.

yy The year, represented as the last two digits of the year, e.g., "09".

19. ACRONYMS

| | |
|---------|---------------------------------------|
| AAG | <i>Athena Analyst's Guide</i> |
| AAM | Athena Attrition Model |
| ALOS | Actual Level of Service |
| AUG | <i>Athena User's Guide</i> |
| AUT | Autonomy |
| CGE | Computable General Equilibrium |
| CLI | Command Line Interface |
| CMO | Civil Military Operations |
| CPI | Consumer Price Index |
| CUL | Culture |
| DAM | Driver Assessment Model |
| DGDP | Deflated Gross Domestic Product |
| ELoS | Expected Level of Service |
| ENI | Essential Non-Infrastructure services |
| EQ | Equal to |
| GDP | Gross Domestic Product |
| GE | Greater than or equal to |
| GT | Greater than |
| GRAM | Generalized Regional Attitude Model |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| HUMINT | Human Intelligence |
| ID | Identifier |
| IGO | Inter-Governmental Organizations |
| JNEM | Joint Non-kinetic Effects Model |
| LE | Less than or equal to |
| LT | Less than |
| LOS | Level of Service |
| MAD | Magic Attitude Driver |
| MAG | <i>Mars Analyst's Guide</i> |
| MAM | Mars Affinity Model |
| MIL-STD | Military Standard |
| NGO | Non-Governmental Organizations |
| PCF | Production Capacity Factor |
| QOL | Quality of Life |
| RDB | Run-time Database |
| RGB | Red-Green-Blue |
| RLOS | Required Level of Service |
| ROE | Rules of Engagement |
| S&RO | Stability and Recovery Operations |
| SFT | Safety |
| SQL | Standard Query Language |

| | |
|--------|--|
| TBD | To Be Determined |
| TCL | Tool Command Language |
| TOE | Table of Organization and Equipment |
| TRADOC | U.S. Army Training and Doctrine Command |
| TRISA | Tradoc G-2 Intelligence Support Activity |
| URL | Universal Resource Locator |

20. SCENARIO CHECKLIST

When creating a scenario, follow the steps outlined in Section 12. These steps are summarized here for convenience. They need not be completed in this precise order, but every step should be considered.

- 1) Define each actor
 - a) What financial resources does the actor have?
 - b) What personnel assets does the actor have?
 - c) How does the actor use his political support?
- 2) Select a map (if desired)
- 3) Define each neighborhood
 - a) Omit irrelevant neighborhoods
 - b) Scale neighborhoods according to your needs
 - c) Neighborhood is local or non-local
 - d) Assign proximities based on the social/psychological distance between the neighborhoods
- 4) Define each civilian group
 - a) Neighborhood of residence
 - b) Population
 - c) Consider belief systems when breaking the population into groups
- 5) Define each force group
 - a) Owning actor
 - b) Uniformed or not
 - c) Set group costs
- 6) Define each organization group (if any)
 - a) Owning actor
 - b) Set group costs
- 7) Define the belief systems of the actors and civilian groups. This is likely to be an iterative process.
 - a) Topics
 - b) Beliefs
 - c) Sanity check
 - d) Tweak results as needed
- 8) Define satisfaction levels and trends for each civilian group
- 9) Define cooperation levels and trends for each civilian group and force group
- 10) Define the status quo deployment of troops across the playbox
- 11) Define the status quo ENI funding for each group by each actor
- 12) Create initial environmental situations (if any)
- 13) Define the actors' strategies
 - a) Put the tactics in priority order
 - b) Create goals to simplify the conditions attached to the tactics
- 14) Make any required changes to the model parameter database