Athena Analyst's Guide

Athena S&RO Simulation, V2.1

November, 2010

William H. Duquette

William.H.Duquette@jpl.nasa.gov

Jet Propulsion Laboratory

Table of Contents

1. INTRODUCTION

This document presents the models and related constructs implemented in version 2.1 of the Athena Stability & Recovery Operations (S&RO) Simulation. The models are described in sufficient detail to allow implementation; neither the derivations nor the the implementation itself are in the scope of this document.

1.1 Athena Spirals

Athena is being developed using the spiral development model. Athena 1.1 was the delivery for Spiral 1; Athena 2.1 is the delivery for Spiral 2A. Athena documentation generally refers to the software version number rather than the spiral.

1.2 Overview

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 is a descendant of the Joint Non-kinetic Effects Model (JNEM), but includes many new models. In addition, where JNEM is a federated simulation, Athena is a stand-alone single-user application.

1.3 Other Documents

Documentation on using Athena may be found in the on-line help; invoke the Athena Simulation, and select Help Contents from the Help menu. Additional documentation may be found in the "docs" directory of the Athena build tree; open "docs/index.html" in a web browser, and follow the links. The documentation is included with the installed software. Documentation can also be obtained directly from the Athena project; contact `William.H.Duquette@jpl.nasa.gov`.

Documentation in the "docs" directory includes:

Athena Rules

This document describes the events and situations (drivers) that affect group attitudes, and the Driver Assessment Model (DAM) rule sets that assess attitude change.

Mars Analyst's Guide

Athena is built upon a software infrastructure layer called Mars. Models implemented by Mars, including the Generalized Regional Attitude Model (GRAM), are documented in the Mars Analyst's Guide (MAG), which may be found in the Athena documentation tree.

Software Manual Pages

Extensive documentation of the Athena software tools and libraries is included in the software installation set in the form of software “man pages”.

1.4 Changes for Athena 2.1

The following changes were made to the Athena models for version 2.1, by major modeling area:

Ground Model: The Ground model underwent a significant paradigm shift with regard to the representation of personnel and population. Previously, force and organization group personnel were represented by units; the analyst could create units and move them about at any time. Group activities were implemented by assigning activities to specific units. The difficulty with this approach is that it becomes hard to manage activity changes during one-to-three-month game turns. Thus, the following changes were made:

- Force and organization group personnel are now explicitly stationed in particular neighborhoods, much as civilians reside in neighborhoods. These force levels can be modified over time.
- Activities are controlled by an activity schedule; each "morning", the activities for the day are staffed, in priority order, from the available personnel. This applies to civilian personnel as well. Staffed activities are represented as units.
- There is no longer as strong a distinction between civilian population and civilian personnel. All people are represented both implicitly (as population or a force stationed in a neighborhood) and as units.

As part of this change, we've eliminated much of the asymmetry between civilian groups and the other group types, which results in simplifications across Athena, and particularly in the Athena Attrition Model (AAM).

Economics Model: The Economics model is entirely new in Athena 2.1.

Demographics Model: Changes here were primarily in response to the needs of the other models. In particular:

- Because of the Ground model changes, the old distinction between "implicit" and "explicit" population is gone, and so the bookkeeping the Demographics model did as civilian units were created, updated, and destroyed has been removed.
- The neighborhood population is now broken down according to the needs of the Economics model. In particular, we divide each neighborhood group into the *subsistence agriculture population*, farmers and herders who are self-supporting and do not participate in the cash economy, and the *consumers* who rely on purchased goods. The *consumers* are then further broken down into workers and non-workers.
- In response to the unemployment rate computed by the Economics model, the Demographics

model creates the **UNEMP** demographic situation, which affects the mood of the population in response to significant unemployment.

Attitudes Model: The Driver Assessment Model rule sets have been revised according to changes made to the JNEM rule sets by the JNEM committee.

1.5 Changes for Athena 2.1.1

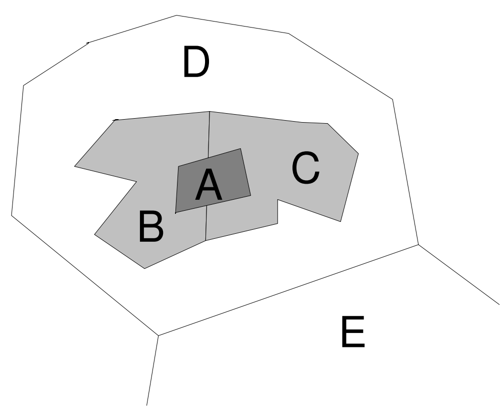
The Economic model's CGE has changed to better account for workers who stay home from work out of fear. Such workers were previously counted as having left the work force; now they are counted as having left their jobs, and hence as being unemployed. This is not entirely satisfactory; Athena 3 will incorporate a more complete solution.

2. ATHENA CONCEPTS

This section gives an overview of Athena and its parts, and of the basic concepts that Athena uses. The discussion is kept to a high level; see sections 3 and following for the detailed models.

2.1 The Athena Playbox

Athena models population dynamics and group activities in a geographical region called the Athena *playbox*. The playbox is divided into areas called *neighborhoods*. Neighborhoods are simply a way of dividing the playbox into a number of reasonably homogeneous areas, and may be of any size: country, province, city, town, zip code, or neighborhood proper. Geographically, neighborhoods are defined as polygons on a map, using some appropriate coordinate system. Neighborhoods may nest, e.g., a city may be a neighborhood within a larger province, and the city may contain several neighborhoods. In the diagram below, A is an urban area surrounded by suburban areas B and C; all three lie within D, a county, which abuts E, another county.



Note that locations within an inner neighborhood are not also part of the outer neighborhood; in the diagram above, D effectively has a hole cut out of it by B and C, and A cuts sections out of both B and C. Consequently, if a neighborhood is completely tiled by nested neighborhoods, it can be omitted as it contains no locations. If D's surface were entirely covered by suburbs, for example, there would be nothing left of D and no reason to define D within Athena.

Although Athena allows neighborhoods that overlap without nesting, like A, it works better in practice if stacked neighborhoods are properly nested. Consequently, the borders of B and C should go around A, rather than A being overlaid on top of them.

In Athena, simulation events take place within neighborhoods, and affect the population of the neighborhoods. An event taking place within a neighborhood can have ripple effects in other neighborhoods; the geographic spread of these ripples depends on how nearby other neighborhoods are presumed to be, not merely geographically but also socially and psychologically. The nearness of one neighborhood to another is an input to Athena called *neighborhood proximity*. There are four proximity levels: *here, near, far,* and *remote*. The above diagram shows proximity to neighborhood A. From A's point of view, A is *here*, B and C are *near* A, and outlying area D is *far* from A. Neighborhood E is *remote*. An event in A would affect A immediately, would likely affect B and C, though to a lesser degree, might affect D to a much lesser degree, and would not affect E at all. Ripple effects in other neighborhoods can also be delayed by an interval, which is an input for each pair of neighborhoods.

2.1.1 Local vs. Non-Local Neighborhoods

All neighborhoods are flagged as being *local* or *non-local*, based on whether or not they participate directly in the economy of the region of interest, also called the *local region*. If the region of interest is Pakistan, for example, we might include portions of India and Afghanistan in the playbox but exclude them from the local economy.

2.2 Groups

The people in the playbox are divided into *groups*, of which there are three kinds: civilian groups, force groups, and organization groups.

2.2.1 Civilian Groups

Civilian groups represent the population of the playbox, i.e., the people who actually live in the neighborhoods. This population may be broken into groups by ethnicity, religion, language, social class, political affiliation, or any other demographic criteria the analyst deems necessary. Civilian groups are similar to the “market segments” used to target advertising: a group is a collection of people who may be assumed to have similar biases, interests, and behaviors due to their demographic similarity.

Athena tracks the civilian population by group and neighborhood; each neighborhood must have a non-zero population of at least one of the civilian groups, and may include representatives of all of the civilian groups. A civilian group residing in a specific neighborhood is referred to as a *neighborhood group*.

Each neighborhood group’s population is represented implicitly in the Demographic model, broken down in various ways: the total population, the number of consumers, the number of workers, the subsistence agriculture population, and so forth. In addition, all civilians are also represented in the Ground model in the form of civilian units. People present in units are usually referred to as *personnel*, rather than as *population;* in Athena, however, unlike JNEM, there is no real difference between the two.

Athena models neighborhood groups in detail, tracking the attitudes of each group in each neighborhood as the group’s members are affected by a variety of events and situations.

2.2.2 Force Groups

Force groups represent military forces, such as the U.S. Army, and other groups whose purpose is to apply force in support of policy. There are five kinds of force group:

- Regular military, e.g., the U.S. Army
- Paramilitary, e.g., SWAT teams and other combat-trained police units
- Police, e.g., normal civilian police
- Irregular military, e.g., militias
- Criminal, e.g., organized crime

The force group type only affects the degree to which a force group’s units are able to project force vs. other force groups. Force group personnel are stationed in particular neighborhoods; we refer to the total number of personnel stationed in a neighborhood as the *force level* in that neighborhood. The troops present in the neighborhood can then be scheduled to engage in a wide variety of activities that affect civilian attitudes, and may engage in armed conflict with other force groups. Scheduled activities are staffed each day; both personnel assigned to activities and the remaining unassigned personnel are represented as units in the Ground model.

A force group may be *uniformed* or *non-uniformed*; this affects the tactics the force group may use. In an S&RO situation, it is assumed that uniformed and non-uniformed forces usually use asymmetric tactics: the uniformed forces (typically regulars, paramilitary, or police) are hunting for cells of non-uniformed forces (typically irregulars), and the non-uniformed forces are attempting to whittle down the uniformed forces by means of IEDs and hit-and-run attacks.

2.2.3 Organization Groups

Organization groups represent organizations that are present in the playbox to help the civilians. There are three kinds: Non-Governmental Organizations (NGOs), International or Inter-Governmental Organizations (IGOs), and Contractors (CTRs). NGOs are groups like the Red Cross or Doctors Without Borders who do humanitarian relief, development, and so forth. IGOs are international organizations like UNESCO. Contractors are commercial firms who are doing development work in the playbox, often but not necessarily working for the Coalition/Government.

Organization group personnel are stationed in neighborhoods and assigned to activities in the same way as force group personnel; activities are staffed each day and both assigned and unassigned personnel are represented as units in the Ground model.

Athena tracks the attitudes of each organization group toward working in each neighborhood; a group that is willing to work in a low-risk neighborhood might be unwilling to work in a neighborhood it perceives as high-risk.

2.2.4 Group Force and Neighborhood Security

Both civilian and force groups, and to a much lesser extent organization groups, have the ability and willingness to project and use force. Athena analyzes the balance of forces in each neighborhood, taking into account the populations and personnel present in the neighborhood, the types of each, and the ability of each group and unit type to project force. As the result of this analysis Athena computes the *security* of each group in each neighborhood. A force or organization group's security in a neighborhood will determine which activities it can perform, if any.

2.3 Modeling Areas

Athena's models are loosely grouped in a number of areas. The models themselves will be described in detail in the body of this document; this section describes each area and the models within it at a high level.

2.3.1 Ground

The most basic area is the Ground model. It includes the neighborhoods, groups, and units, as described above, and also the following specific models:

- Force security
- Group activity scheduling, staffing, and analysis, including Activity Situations (actsits).
- The Athena Attrition Model (AAM)
- Environmental Situations (ensits)

2.3.2 Demographics

The Demographics model tracks the civilian population of the playbox by neighborhood and group, and breaks down each neighborhood group's population in a variety of ways. Populations change as civilians lives are lost due to collateral damage and direct attrition, and as civilians are displaced to other neighborhoods. The Demographics model provides population statistics, e.g., the number of consumers and the size of the labor force, to the Economics model, and creates demographic situations (demsits) in response to the rest of Athena.[1]

1 In Athena 2.1 there is only one demsit type, Unemployment; see the *Athena Rules* document for details.

2.3.3 Attitudes

The Attitudes model is responsible for tracking the effects of events and situations (collectively known as *drivers*) on the attitudes of neighborhood and organization groups. Attitudes currently include satisfaction of a group with respect to a variety of concerns, and cooperation of neighborhood groups with force groups.

The engine responsible for tracking attitudes and changes to attitudes is called the Generalized Regional Attitude Model (GRAM); it is documented in great detail in the *Mars Analyst's Guide*, which is delivered with Athena; see section 1.3 .

The *Driver Assessment Model* (DAM) is responsible for assessing the effects of each driver of attitude change, and giving related inputs to GRAM. DAM primarily consists of a large collection of rule sets; each rule set is devoted to one particular kind of driver, e.g., civilian casualties or presence of a force group. The DAM rule sets are described in the *Athena Rules* document; see section 1.3 . A wide variety of drivers already exist: activity situations, environmental situations, demographic situations, and civilian and organization casualties; in addition, the user may created their own *Magic Attitude Drivers* (MADs), in effect writing DAM rules on the fly.

2.3.4 Economics

The new Economics model tracks employment and the production of goods and services in the local region. The economy changes in response to changes in neighborhood demographics and production capacity, and (via the Demographics model) drives the new Unemployment situation.

2.4 Simulated Time

Athena uses the following measures of simulated time.

Athena's clock measures time in integer *ticks* since time 0. The tick is the smallest time interval with which Athena is concerned; simulation time always advances tick-by-tick. Each tick represents a single day of simulated time.

Time 0 is mapped to a calendar date by a *start date* set by the user. Athena then outputs simulated time as either some number of integer ticks or as a zulu time string based on the start date.

Athena is a time-step simulation; many computations take place at each time tick. Some models are triggered every so many ticks; these trigger points are called *tocks*. For example, economics and attrition are assessed every seven ticks (by default); thus, we say that the economics tock and the attrition tock are each seven ticks.

We expect the analyst to run Athena forward in increments of 30 to 90 days (ticks), pausing at the end of each interval to check the status and make changes. These increments are sometimes referred to as *game turns*, or just *turns*.

2.5 Simulation States and the Advancement of Time

The Athena simulation has a number of states. When a new scenario is created, Athena is in the Scenario Preparation or **PREP** state. In this state, the analyst creates neighborhoods, groups, and so forth, and sets up their initial attributes.

When the analyst is satisfied with the scenario, the scenario is *locked*. At this time, Athena:

- Initializes GRAM.
- Staffs activities based on the initial activity schedule.
- Computes demographics and force security.
- Calibrates the economics model.
- Executes any simulation orders scheduled to be executed at time 0.
- Enters the **PAUSED** state.

In the **PAUSED** state the analyst may set force levels, schedule force level changes, schedule activities, and execute and schedule many other simulation orders. Then, the analyst may request a *game turn* of from 1 tick to 90 ticks in length. The simulation state changes to **RUNNING**, and time advances tick by tick.

During each tick, Athena performs the following steps:

- Updates the demographics and activity statistics.
- Assesses attrition and the economy (if it is the correct tock)
- Assesses environment, activity, and demographic situations.
- Advances GRAM.
- Saves a variety of historical data, for later plotting.
- **Updates the simulation time by one tick.**
- Executes any scheduled events:
  - Spawning or resolving of environment situations
  - Scheduled orders
- Staffs activities.

- Returns to the **PAUSED** state if the stopping time has been reached, allowing the analyst to make changes.

It is useful to think of these various computations as being performed at different times of day:

- Midnight: the simulation time is advanced by one tick.
- Morning: activities are staffed and scheduled orders are executed.
- Noon: if **PAUSED**, the analyst can make changes interactively.
- Evening: the activities and events of the day are assessed for their attitude implications, and GRAM is advanced.
- Midnight: the simulation time is advanced by one tick.

In short, the simulation pauses in the middle of the tick. As a result, if a situation begins at time t the user will first see it at time $t+1$.

3. FORCE ANALYSIS

Athena analyzes the forces present in each neighborhood and the neighborhood demographics, to determine the following quantities for each group and neighborhood:

Security: A group level of security in a neighborhood, measured as a score from -100 (very dangerous) to +100 (very safe). A group's security determines the kinds of activities the group can safely conduct within the neighborhood.

Volatility: The volatility of a neighborhood is the likelihood of spontaneous violence within the neighborhood, measured as a score from 0 (least volatile) to 100 (most volatile).

Force: A group's force in a neighborhood is its physical ability to use its assets to control that neighborhood through force. The major component of force is military strength.

3.1 Measuring Force

In general:

- The primary component of group *g*'s force in neighborhood *n* is group *g*'s assets in the neighborhood. In Athena, a group's assets are its people. Athena has no notion of equipment.
- Friendly military personnel in neighborhood *n*, i.e., those with a positive relationship with *g*, should increase *g*'s force in proportion to their numbers.
- Friendly civilians in *n* should also increase *g*'s force, but by a smaller amount per person than soldiers or militia. However, an aggressive demeanor and a negative mood (S_{ng}) can increase a civilian group's force multiplier.
- Friendly contractor personnel in *n* may also increase *g*'s force, but personnel from friendly IGOs and NGOs are expected to remain neutral in case of conflict and do not contribute to *g*'s force.
- Friendly military and civilian personnel in nearby neighborhoods should also increase *g*'s force, but to a smaller extent than military and civilian personnel in neighborhood *n*.
- Also of interest is the force aligned against group *g*, from the groups that oppose group *g*. And just as *g* can call in its friends from nearby neighborhoods, so can *g*'s enemies.

Using these concepts we will build up a notion of force which can be used to define neighborhood volatility, and then a group's security within the neighborhood.

First, group *g*'s own force in neighborhood *n* is

$$Q_{ng} = \text{ceiling}\left(F_{ng} \times personnel_{ng}\right)$$

where

F_{ng}	=	The force multiplier for group *g* in neighborhood *n*. This value is defined differently for civilian groups than other groups.
$personnel_{ng}$	=	The number of group *g*'s personnel present in neighborhood *n*.

Force Multiplier for Civilian Groups: For civilian group *g* in neighborhood *n*, the force multiplier F_{ng} is defined as follows:

$$F_{ng} = a \times D_{ng} \times M_{ng}$$

Here, *a* is a multiplier, nominally 0.1, which reflects that only a small fraction of the population will be available to participate in a fracas at any given time.[2] D_{ng} is a multiplier based on the demeanor of *g* in *n*.[3] M_{ng} is a multiplier based on the mood S_{ng} of group *g* in *n*. It is assumed that the more aggressive the demeanor and the worse the mood, the more likely it is that civilians will use force to aid their friends and hinder their foes:

$$D_{ng} = \begin{cases} 1.5 & \text{if demeanor is Aggressive} \\ 1.0 & \text{if demeanor is Average} \\ 0.3 & \text{if demeanor is Apathetic} \end{cases}$$

$$M_{ng} = \left(1 - b \cdot \frac{S_{ng}}{100}\right)$$

Here, *b* is a factor that determines how strongly a group's mood contributes to its force.[4] If *b* is 0.2, for example, the effect of mood will range from 0.8 (when group *g* is perfectly satisfied) to 1.2 (when group *g* is perfectly dissatisfied).

Force Multiplier for Force and Organization Groups: For force or organization group *g* in neighborhood *n*, the force multiplier F_{ng} is intended to account for the effectiveness and demeanor of the unit's personnel:

$$F_{ng} = E_g \times D_{ng}$$

D_{ng} is defined above; note that demeanor is constant across neighborhoods for force and organization

2 Model parameter: `force.population`
3 Model parameter: `force.demeanor.*`
4 Model parameter: `force.mood`

groups. E_g, the unit force multiplier, should ideally depend on the type of the unit and the nature of its assets. In Athena, however, all units are the same and have no assets other than personnel, and so E_g is constant for each group. It is defined as shown in the following table:[5]

Characteristics of Unit *i*			E_g
Unit's group type is Organization (ORG)	Group's ORG type is:	NGO	0
		IGO	0
		Contractor	2
Unit's group type is Force (FRC)	Group's force type is:	REGULAR	25
		PARAMILITARY	15
		POLICE	10
		IRREGULAR	20
		CRIMINAL	8

Effect of unit activities: At the January 2007 meeting of the JNEM Rules Committee, it was suggested that a unit's activity could affect its force multiplier. Although this is an intuitively attractive notion, it may contain an implicit circularity. Our model presumes that a unit can successfully perform an activity given sufficient security. If the unit's extra contribution to force due its activity depends on its ability to successfully perform the activity, then we need to have computed security to determine the unit's contribution to force...yet we need the unit's contribution to force in order to compute security.[6] Consequently, the committee ruled that a unit's activity need not affect its force.

Effect of Group Relationships: Friends and enemies are determined by their Group Relationship, R_{nfg}, where friends have a positive relationship and enemies have a negative relationship. We assume that group *f* will assist or oppose group *g* according to the strength of the relationship from *f*'s point of view. For convenience we define

$$R^{+}_{nfg} = \begin{cases} R_{nfg} & \text{where } R_{nfg} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$R^{-}_{nfg} = \begin{cases} |R_{nfg}| & \text{where } R_{nfg} < 0 \\ 0 & \text{otherwise} \end{cases}$$

5 Model parameters: `force.orgtype.*`, `force.forcetype.*`

6 RGC: The egg had to come before the chicken. That is, whether the unit has enough security to conduct an activity has to be based on the security before it started doing the activity or it could not have started. However, rather than introduce a new time lag (which we could do), I recommend we merely look at the previous value. With this rationalization, it is no longer circular, but inevitably sequential. It would become circular if the activity lowered the unit's effectiveness. I claim that should be taken into account when establishing the security threshold for that activity.

Then,

$$LocalFriends_{ng} = \sum_{f} Q_{nf} \cdot R^{+}_{nfg}$$
$$LocalEnemies_{ng} = \sum_{f} Q_{nf} \cdot R^{-}_{nfg}$$

Note that R_{nfg} need not be symmetric; if we were to replace R_{nfg} with R_{ngf} in $LocalFriends_{ng}$, then we'd be assuming that group *f* would assist group *g* based on the strength of *g*'s feeling for *f*, which might not be at all the same thing as *f*'s feeling for *g*.

Next, we take neighborhoods into account:

$$Force_{ng} = LocalFriends_{ng} + h \cdot \sum_{m \text{ near } n} LocalFriends_{mg}$$
$$Enemy_{ng} = LocalEnemies_{ng} + h \cdot \sum_{m \text{ near } n} LocalEnemies_{mg}$$

Here, *h* is a factor, nominally 0.3, that reduces the effect of friends and enemies in nearby neighborhoods.[7] The phrase "*m* near *n*" denotes those neighborhoods *m* whose $Proximity_{mn}$ is *near*, that is, those neighborhoods *m* whose inhabitants regard neighborhood *n* as being nearby. As with R_{nfg}, $Proximity_{mn}$ need not be symmetric.

In order to create scores, we'll need to normalize these values by the total force in the neighborhood:

$$TotalForce_{n} = \sum_{g} Q_{ng} + h \cdot \sum_{m \text{ near } n} \sum_{g} Q_{mg}$$

Then,

$$\%\,Force_{ng} = \frac{Force_{ng}}{TotalForce_{n}} \times 100$$
$$\%\,Enemy_{ng} = \frac{Enemy_{ng}}{TotalForce_{n}} \times 100$$

7 Model parameter: `force.proximity`. An appropriate value of *h* should be selected during exercise construction to reflect the criteria used for selection of neighborhood boundaries. Normally, a playbox with large neighborhoods would have a smaller value for *h* than one with small neighborhoods.

3.2 Volatility

The volatility of a neighborhood is the likelihood of spontaneous violence within the neighborhood. Volatility depends on the balance of forces in the neighborhood, and is a key component of security. For example, an ORG group may hesitate to go into a neighborhood with high volatility, even if the ORG group is on friendly terms with all of the parties present in the neighborhood, simply because the chance of getting caught in a cross-fire is too high.

The volatility of neighborhood *n* can be regarded as a measure of the number of individuals potentially in conflict in neighborhood *n*, using the force values defined above:

$$Conflicts_n = \sum_g Enemy_{ng} \times Force_{ng}$$

$Conflicts_n$ ranges from 0 to $TotalForce_n^2$ because it counts every conflict from the point of view of each of the parties involved. Scaling to the range 0 to 100 yields

$$Volatility_n = \frac{Conflicts_n}{TotalForce_n^2} \times 100$$

3.3 Security

A group's security in a neighborhood determines the kinds of activities the group can perform within that neighborhood. Different levels of security are needed for different kinds of operations. The group's level of security depends on the assets present in the neighborhood, including both those belonging to the group and those belonging to friendly and unfriendly groups, as well as on the nature of both assets and groups and the general volatility of the neighborhood. In general:

- The presence of friendly forces should increase group *g*'s security within neighborhood *n*.
- The presence of enemy forces should decrease group *g's* security within neighborhood *n*.
- Increased volatility should decrease group *g*'s security within neighborhood *n*.

The security of group *g* in neighborhood *n* is defined as follows:

$$Security_{ng} = \frac{\%Force_{ng} - \%Enemy_{ng} - v \cdot Volatility_n}{100 + v \cdot Volatility_n} \times 100$$

where *v* is the volatility scaling factor, nominal 1.0, which can be used to reduce the effects of volatility on security.[8] The denominator scales $Security_{ng}$ so that it ranges from -100 to +100.

8 Model parameter: `force.volatility`

For use in rules, $Security_{ng}$ will usually be translated to a symbolic value, as shown in the following table:

Range	Symbol
$25 < Security_{ng} \leq 100$	High
$5 < Security_{ng} \leq 25$	Medium
$-25 < Security_{ng} \leq 5$	Low
$Security_{ng} \leq -25$	None

4. ACTIVITY SCHEDULING AND STAFFING

The activities performed by a force operating in a region can significantly affect the attitudes of the local civilian population; and the manner in which the activities are carried out in the short run can ease or obstruct the commander's job in the long run. This section explains how activities are scheduled and staffed with personnel in Athena; Section 5. lists the different activities and explains how their effects on the local civilians are determined.

4.1 Assigned vs. Coincidental Activities

There are two kinds of activities: assigned and coincidental. Assigned activities are simply those that can be assigned to personnel, such as patrolling. Coincidental activities are unassigned but unavoidable given the group's presence and assigned activities. For example, a force group cannot patrol a neighborhood without being present in the neighborhood; and the mere fact of the group's presence affects the civilians, independent of what the group's members are doing.

This section is concerned only with assigned activities.

4.2 Staffing Pools

Assigned activities have to be performed by group personnel, and those personnel must come from somewhere. Thus, we have the notion of a *staffing pool*, a body of personnel resident in a neighborhood that can be assigned activities.

For civilian groups, the staffing pool is simply the group's population in the given neighborhood. This is set during scenario preparation and does not increase as simulation time advances, though it may decrease due to attrition.

For force and organization groups, the analyst must station personnel in neighborhoods; this is called setting or adjusting the group's force level in a neighborhood. The group's personnel stationed in a neighborhood then become a staffing pool, and can staff a variety of activities in the same or other neighborhoods.

The neighborhood in which personnel are stationed is called the *neighborhood of origin* for those personnel. The neighborhood in which personnel assigned an activity carry out that activity is called the *neighborhood of operation*.

4.3 Scheduling and Staffing Activities

In Athena 1.0, the analyst could create units on the fly, move them from neighborhood to neighborhood, and assign them activities directly. Once positioned, units remained in position, performing (or attempting to perform) their activities, until the analyst chose to move them.

This mode of interaction derives from the training environment in which many of Athena's models were initially developed:

- The simulation runs as fast as it runs
- As it runs, the analyst enters orders to units
- The analyst watches what happens as it happens.

The use case for Athena, on the other hand, involves running one-to-three month "game turns". The analyst is supposed to set up a course of action, let it run the game turn on its own, and then see what happened. Athena 1.0 supported this use case only if the analyst didn't want anything to change during the game turn. But for long game turns, this simply isn't realistic.

Consequently, Athena 2.1 allows the analyst to think in terms of planning force levels and scheduling activities. The schedule may be arbitrarily complex; and each day the available forces will be used to staff the scheduled activities. Thus, complex courses of action can be carried out during a game turn.

4.4 Scheduling Activities

The activity schedule consists of a series of calendar items *i*, each of which has the following attributes:

g_i	=	The group performing the activity
n_i	=	The *neighborhood of origin* for the personnel performing the activity, i.e., the neighborhood in which they are stationed or reside.
a_i	=	The activity to be performed
tn_i	=	The neighborhood in which the activity is to be performed, also known as the *target neighborhood* or the *neighborhood of operation*.
$personnel_i$	=	The number of personnel scheduled to perform the activity.
$start_i$	=	The first day (specified as an integer tick) on which the activity might be scheduled.
$finish_i$	=	The last day (specified as an integer tick) on which the activity might be scheduled. $finish_i$ can have the special value "NEVER", indicating that the activity will continue indefinitely. The span from *start* to *finish* is called the scheduled interval.
$pattern_i$	=	A pattern indicating the days in the scheduled interval upon which the activity is actually scheduled (see below).
$priority_i$	=	This calendar item's priority, from 1 (highest) to N. Calendar items are staffed in priority order.[9]

The subscript *i* will be omitted when confusion is unlikely.

9 It is assumed that the application will show priority by position in a vertical list; hence, no two calendar items will have the same priority.

Calendar Patterns: The value of *pattern* determines the days upon which the activity is actually scheduled. In a typical calendar application, a meeting might be scheduled to occur on the first and third Thursdays of every month from September to June; September 1st is the *start*, June 30th is the *finish*, and "the first and third Thursday" is the *pattern*. Athena provides the following patterns:

daily	The activity is scheduled to occur on every day during the interval, and will be restaffed each day, personnel permitting. Personnel lost due to attrition will be replaced.
once	The activity is scheduled to occur on every day during the interval, but is staffed only once, at the beginning of the interval. Personnel lost due to attrition are not replaced.
byweekday	The activity is scheduled to occur on a specified set of week days, e.g., Monday, Wednesday, and Friday, and will be staffed on each of these days during the interval, personnel permitting.

It is likely that other patterns will be added over time.

Duplicate Activities: Note that the same activity can be scheduled multiple times on any given day; available personnel permitting, all such are fully staffed. If the analyst wants troops patrolling every day but additional troops patrolling on Fridays, for example, he can schedule one calendar item that provides the base level of troops every day and another that provides additional troops on Fridays.

4.5 Staffing Activities

Activities are staffed "first thing in the morning", immediately after time advances to a new tick. Staffing for group *g* in neighborhood *n* is done according to the following algorithm.

Let *pool* be the number of group *g*'s personnel in its staffing pool in neighborhood *n*:
Let *t* be the current simulation time.
For each calendar item *i*, in order from highest priority to lowest, that is active at time *t*,
    If the *pattern* is **once** and $t \neq start$,
        This activity has already been staffed.
        Go on to the next item.
    Let $staff = min(personnel, pool)$
    Let $pool = pool - staff$
    If *staff* is 0,
        The activity cannot be staffed.
        Terminate staffing.
    Deploy a unit belonging to group *g* with neighborhood of origin *n* and neighborhood of operation *tn*, containing *staff* personnel and performing activity *a*.
If *pool* > 0,
    Deploy a unit containing *pool* troops doing activity **NONE** in the neighborhood of origin.

4.6 Activities and Units

In Athena 2.1, the analyst does not create units; rather, he schedules activities. Athena then automatically creates a unit for each calendar item when the calendar item's activity is first staffed.

A unit's position within a neighborhood is irrelevant to the simulation. Newly-created units are positioned randomly in their neighborhood of operation; the analyst may then position them anywhere in the neighborhood, as an aid to visualization.

Unassigned personnel--those who remain in the staffing pool after all of the day's activities are staffed--are placed in a *base unit*: a unit located in the pool's neighborhood of origin and having activity NONE. Once created, base units remain indefinitely, though they might often contain zero personnel. Thus, in Athena 2.1 all personnel, whether force, organization, or civilian, are represented in units.

5. EFFECTS OF UNIT ACTIVITIES

Given that activities have been scheduled and units have been staffed, as described in Section 4., the units and their activities will affect the civilian population as described in this section.

5.1 Force Presence and Activities

Force units present in a neighborhood can affect civilian attitudes in a neighborhood in two ways: by their mere presence, and by engaging in other activities.

Force Presence: A force group's presence in a neighborhood is measured as the total number of personnel in units that belong to the group and are present in the neighborhood.[10]

Force Activities: A unit assigned an activity is assumed to carry out the activity if it can possibly do so. Each activity requires that the acting group have a minimum level of security in the neighborhood. The following table lists the force activities, and the nominal minimum security level for each.[11]

Force Activity	Minimum Security Required	Shifts	Coverage Function
Presence	n/a	1	25/1000
Checkpoint	Low	1	25/1000
CMO -- Construction	High	1	20/1000
CMO -- Development (Light)	Medium	1	25/1000
CMO -- Education	High	1	20/1000
CMO -- Healthcare	High	1	20/1000
CMO -- Industry	High	1	20/1000
CMO -- Infrastructure	High	1	20/1000
CMO -- Law Enforcement	Medium	1	25/1000
CMO -- Other	High	1	20/1000
Coercion	Medium	1	12/1000
Criminal Activities	Medium	1	10/1000
Curfew	Medium	1	25/1000
Guard	Low	1	25/1000

10 Ideally, the nature of each unit would also be taken into account, at least at the level of "boots, wheels, or tracks," i.e., the number of personnel on foot, in wheeled vehicles, and in tracked vehicles. Tanks tear up the pavement, for example, and thus should have an increased negative effect on QOL. At present, Athena does not model units to this level of detail.

11 Model parameter database, `activity.FRC.*`

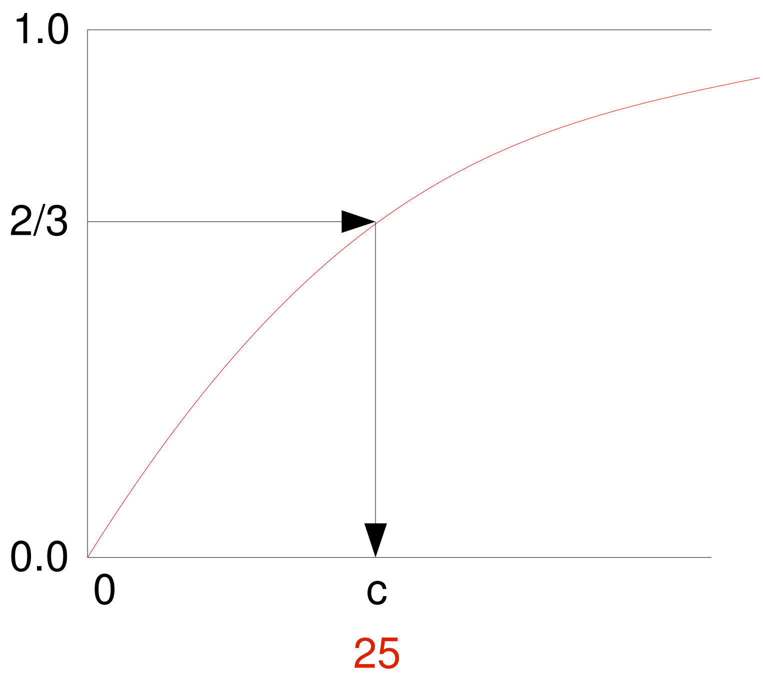
Force Activity	Minimum Security Required	Shifts	Coverage Function
Patrol	Low	1	25/1000
PSYOP	Medium	1	1/50000

Thus, a force group can enforce a curfew or provide law enforcement in a neighborhood only if the group’s security is at least Medium in that neighborhood. The extent to which a group is engaging in an activity in a neighborhood is measured by the total number of personnel in units that:

- Belong to the group
- Are present in the neighborhood
- Are assigned to do the activity
- Are on the current shift (the number of personnel available at any given time is the total divided by the number of shifts defined for the activity type)

This figure is automatically 0 if the minimum security requirement is not met.

Coverage Fractions: Whenever a group is present or conducting an activity in a neighborhood, the extent to which its presence or activities affect the local population is termed its *coverage fraction* for the activity. The coverage fraction is a value from 0 to 1; it is computed by comparing the group’s personnel (present, or conducting an activity) with the total population of the neighborhood, using a function like that shown below.



In this example, if the group has deployed 25 troops per 1000 people in the neighborhood then the coverage fraction is 2/3, rising exponentially toward 1.0 as the troop density goes toward infinity. The coverage fraction determines how much of the potential satisfaction change is gained or lost by the group's presence or activity. The x-axis is usually troops per 1000 population, as shown, but can vary; for PSYOP, it is troops per 50,000 population. Thus, the two parameters required to compute a coverage fraction are {*c*, *d*} where *c* is the troop density at the 2/3rds point, and *d* is the population denominator, e.g., 1000 or 50,000. The coverage function is often specified in the form *c*/*d*. In the example shown above, then, the coverage function may be specified as either 25/1000 or {25, 1000}. The actual function is as follows. First, the troop density is

$$TD = \frac{p \cdot d}{Population_n}$$

where *p* is the number of personnel engaged in the activity. Then the coverage fraction is

$$CF = 1 - e^{-\frac{TD \cdot \ln 3}{c}}$$

Example: Force group *g* has 750 troops in neighborhood *n*, which has a total population of 40,000 people:

$p = 750$

$Population_n = 40{,}000$

The coverage fraction parameter, *c/d*, is nominally 25/1000 for mere presence. The troop density for group *g* is therefore

$$TD = \frac{p \cdot d}{Population_n} = \frac{750 \cdot 1{,}000}{40{,}000} = 18.75$$

The coverage fraction is therefore

$$CF = 1 - e^{-\frac{18.75 \cdot \ln 3}{c}} = 1 - e^{-0.824} = 1 - 0.44 = 0.56$$

Details of the various presence and activity rule sets are contained in the *Athena Rules* document.

A force group *g*'s coverage fraction for activity *a* in neighborhood *n* is denoted $Coverage_{nga}$.

Composite Coverage Fractions: If two force groups were cooperating in enforcing a curfew, we would expect the curfew to suppress civilian activities based on the total personnel used in enforcing it, independent of the fact that two groups are involved. In cases like these it is useful to employ a *composite coverage fraction*. This fraction can be computed based on the total troop density across all relevant groups; it can also be computed from each group's coverage fraction for the activity, as follows:

$$Coverage_{na} = 1 - \prod_{g} (1 - Coverage_{nga})$$

5.2 Organization Activities

Units belonging to organization groups have no effect on civilian satisfaction due to their mere presence. Organization units may perform the following activities; the effect of these activities is modeled in the same way as force activities are (section 5.1), with these distinctions:

- The minimum security required for an activity depends primarily on the organization type, and not on the activity. NGOs and IGOs require “high” security, while CTRs require “medium” security.[12]
- Organization units can perform only a subset of the activities that a force unit can perform.

Organization Activity	Minimum Security Required	Shifts	Coverage Function
CMO -- Construction	High/Medium	1	20/1000
CMO -- Education	High/Medium	1	20/1000

12 Model parameter: `activity.ORG.*`

Organization Activity	Minimum Security Required	Shifts	Coverage Function
CMO -- Healthcare	High/Medium	1	20/1000
CMO -- Industry	High/Medium	1	20/1000
CMO -- Infrastructure	High/Medium	1	20/1000
CMO -- Other	High/Medium	1	20/1000

5.3 Civilian Activities

Civilian units may perform the following activities; the effect of these activities is modeled in the same way as force activities are (section 5.1), with the distinction that (for the currently defined activities) security is irrelevant.

Civilian Activity	Minimum Security Required	Shifts	Coverage Function
Displaced Persons	n/a	1	25/1000
In Camp	n/a	1	n/a

Units assigned the "Displaced Persons" activity are assumed to contain population driven from their homes and living with and among the civilian population in the area in which they find themselves. A coverage is computed for these persons in the normal way.

Units assigned the "In Camp" activity are assumed to be displaced persons who have been settled in a refugee camp. Such persons are ignored, for two reasons. First, such camps are often established in out of the way places, where there are no local civilians to be affected. Second, the attitudes of the displaced persons themselves are of interest; but GRAM does not yet allow populations to move around in this way.

5.4 Activity Situations

When a group has an activity with coverage greater than 0.0 in a neighborhood, we say that an *activity situation* exists in that neighborhood. The effects of these situations on local attitudes are assessed by the DAM rules; see the *Athena Rules Document*. Note that no activity situation is created for the "In Camp" activity.

6. ENVIRONMENTAL SITUATIONS

Environmental situations are on-going circumstances in a neighborhood that affect all of the civilians resident in the neighborhood. Environmental situations have a coverage fraction, just as activity situations do; however, for environmental situations the coverage fraction is set by the analyst when the situation is created. The complete set of environmental situation types (and the accompanying DAM rules) are given in the *Athena Rules* document; see section 1.3 .

7. ATHENA ATTRITION MODEL (AAM)

The Athena Attrition Model (AAM, pronounced “aim”) models attrition to force, organization, and civilian units. In Spiral 2A, AAM provides a basic framework for attrition and attrition-related modeling that includes most areas, but models each area in a simple way that can be improved in later years.

Unlike traditional, detailed attrition models, which model individual firefights between opposing, detected, uniformed forces at specific points of time at specific locations, AAM models attrition caused by both uniformed and non-uniformed forces over longer periods of time (typically a week) at unspecified—but nearby—locations.

By uniformed forces we are referring to military, paramilitary or other forces wearing uniforms or other outward indication that they belong to a particular force group. By non-uniformed forces we are referring to combatants (typically insurgents) who belong to a particular force group, but who are dressed to blend in with the local civilian population in order to avoid being recognized by enemy forces.

AAM does not explicitly consider the effects of equipment and weapons systems in use by force units (nor are these modeled in Athena). Rather, it is assumed that forces are appropriately equipped for their activities. The effects of equipment and weapons systems are implicit in the parameters of the various equations and algorithms.

In addition, AAM supports magic attrition of individual units, of any specific group in a neighborhood, and of all civilians in a neighborhood.

7.1 Overview

7.1.1 Attrition in the Real World

Attrition is the death of unit personnel due to inter-group violence, ranging from chance altercations to targeted attacks (assassinations and ambushes) to riots to open force-on-force combat.

In the real world, attrition can be caused by uniformed forces, non-uniformed forces, certain types of organizations (ORGs), hostile/militant civilians who do not belong to any force group, and complex crowds containing various mixes of people: non-uniformed forces, civilian supporters of force groups, "rent-a-crowds," and ordinary folks who are sympathetic to the cause, just looking for some excitement, or just happen to get caught up in things. Attrition can be suffered by all of the above, as well as by peaceful organizations and by innocent civilian bystanders.

We do not attempt to deal with all of these possibilities in Spiral 2A.

7.1.2 Requirements for This Spiral

- Attrition shall take place in neighborhoods, based on the groups present in the neighborhood.
- All personnel are present in neighborhoods as unit personnel, i.e., as visible unit icons. staffed according to the activity schedule and personnel available in the relevant staffing pool.
- Force groups may be designated as *uniformed* or *non-uniformed*.
- Crowds of civilians (simple or complex) will not be represented.[13]
- For convenience,
    - A Uniformed Force (UF) is the collection of all units within a neighborhood that belong to a particular force group that is designated as uniformed.
    - A Non-uniformed Force (NF) is the collection of all units within a neighborhood that belong to a particular force group that is designated as non-uniformed.
    - A Civilian "Force" (C) is the collection of all units within a neighborhood that belong to a particular group in the neighborhood.
    - The attitude changes due to attrition to a displaced civilian unit will be done as though the attrition was incurred in the unit's neighborhood of origin.[14]
    - An Organization "Force" (O) is the collection of all units within a neighborhood that belong to a particular organization group.
- A neighborhood may contain any combination of UFs, NFs, Cs, and Os.
- Groups may be friendly, enemy, or neutral with each other, based on the value of their relationships.
- Attrition occurs during engagements between enemy forces.
- AAM shall model attrition caused by the following types of engagement:
    - UF vs. NF
        - UF hunting down enemy NF.
        - NF ambushing enemy UF.

13 At Athena's timescale, most crowd-related phenomena are events rather than situations.
14 This is not quite right; but the current model for displaced civilians is a stopgap until GRAM can support it properly.

    - Collateral damage to C from all engagements between forces.

- All casualties from attrition will be kills; there will be no computation of wounded.

- All attrition to O will be handled by magic input.

- All mass killings of C by any group will be handled by magic input.

- Coercion of civilians by force groups will be modeled as a force activity, as in JNEM; it improves cooperation if the coercing force has sufficient security and coverage. Explicit killing of C by UF or NF for the purposes of coercion will not be modeled in Spiral 2A.[15]

    - Putting more troops into a neighborhood allows greater opportunity for coercion.

- A UF can hunt down enemy NF in a neighborhood. In this case, the UF will benefit in the following ways if it can increase the cooperation of the population with itself and decrease the cooperation of the population with the NF:

    - Fewer attacks on UF by NF

    - Better loss exchange rates for UF when NF does attack

    - More attacks by UF against NF with less collateral damage to C on each attack.

- To find enemy NF, a UF must have troops present in the neighborhood. Increasing the coverage of the troops will produce more attacks on enemy NF but will also make the UF more vulnerable to attack by NF (up to a point).

7.1.3 Requirements for Later Spirals

- To find enemy NF, a UF must have troops in the neighborhood with activities that expose them to the local population. Increasing the coverage of such activities will produce more attacks on enemy NF but will also make the UF more vulnerable to attack by NF (up to a point).

- NF requires funding/resources in order to conduct attacks against UF.

    - Reducing NF funding/resources reduces the number of attacks that NF can initiate against UF.

15 In future spirals we will add:
- Levels of coercion (mere threatening vs. actual killing)
- Tipping points (NPS coercion model)
- Time to become effective
- Levels of effectiveness based on presence of other forces and their activities
- Explicit attrition to civilians for purposes of coercion

- NF can conduct IED attacks against UF.
- Force groups will be allowed to recruit new members from the militant pools of local civilian groups.
- Individual units might be designated as Uniformed or Non-uniformed, perhaps according to their activity, allowing a single force group to have both UF and NF.
- AAM will model attrition caused by the following types of engagement.
  - C vs. C
  - UF vs. UF
  - NF vs. NF
  - Collateral damage to O from all engagements between forces.

7.1.4 Simplifying Assumptions

In order to keep AAM within the realm of the possible in Spiral One we made the following simplifying assumptions, which are still in force in Spiral 2A:

- NFs will ambush UFs to inflict a few casualties and then run in order to limit their own casualties
- When a UF discovers an NF cell, the UF will attack with overwhelming force and suffer no casualties
- Neither UFs nor NFs will intentionally kill civilians in this model.[16]
- NFs will not explicitly use crowds to promote unrest or otherwise further their agenda.
- UFs will cause collateral damage (the killing of civilian bystanders) when attacking or defending against NFs. NFs will not kill civilians when they attack or are attacked by UF.
- There is no direct UF vs. UF or NF vs. NF conflict within the playbox.

16 Mass killings can be handled via magic input. Killing as a coercion technique will not be explicit in Spiral 2A.

7.2 Uniformed vs. Non-Uniformed Forces

Every force group will have a flag indicating whether it is uniformed or non-uniformed. We expect that regular military, paramilitary, and police groups will usually be uniformed (UF), and irregular military and criminal groups will usually be non-uniformed (NF).

7.3 Units and Unit Activities

Rather than representing a specific body of troops in some location, an Athena unit really represents an allocation of some number of personnel to an activity within the neighborhood, such as PATROL or COERCION. Troops assigned the activity NONE are presumed to be in reserve. At Athena's time scale, it makes no sense to model unit movement or location in any detail. Athena units have a precise location within each neighborhood, but only as an aid to visualization.

Some activities involve more exposure to the local population than others. Increased exposure has a number of effects: the unit is more likely to find enemy units; the unit is more likely to get intel from the local civilians; the unit is more likely to be attacked by enemy units. The following force activities are deemed to involve significant exposure to the local civilians:

CHECKPOINT
CMO_CONSTRUCTION
CMO_DEVELOPMENT
CMO_EDUCATION
CMO_EMPLOYMENT
CMO_HEALTHCARE
CMO_INDUSTRY
CMO_INFRASTRUCTURE
CMO_LAW_ENFORCEMENT
CMO_OTHER
COERCION
CRIMINAL_ACTIVITIES
CURFEW
PATROL
PSYOP

However, the results of such increased exposure are being deferred to a later spiral.

The following activities do not have significant exposure in the sense used here:

GUARD
NONE

7.4 Unit Number and Unit Size

Because units are used to allocate troops to particular activities, rather than to represent any military

TOE, the breakdown of a force into units has no tactical significance. Consequently, the breakdown of a force into units will be ignored when computing the amount of attrition each group incurs.

7.5 Attrition and Staffing Pools

All attrition comes out of the attrited units; but it also comes out of each unit's staffing pool, thus reducing the personnel available for staffing the following day. The following sections will speak only of attrition to units; the effect on the staffing pools is implicit.

7.6 Magic Attrition

The analyst can attrit units and groups magically. All attrition to civilian or organization units will be assessed by the relevant DAM rule set. Attrition to civilians displaced from their neighborhood of origin will be assessed as though they were attrited in their neighborhood of origin.

7.6.1 Magic Attrition to Units

The analyst can attrit a specific unit, of any type.

7.6.2 Magic Attrition to Groups

The analyst can attrit a specific group in a specific neighborhood. If the group is a civilian group, only units resident in the neighborhood will be attrited.

7.6.3 Magic Attrition to Neighborhoods

The analyst can also choose to apply attrition to all civilian units that happen to be a neighborhood; this is equivalent to collateral damage incurred during normal attrition.

7.7 Antagonists and ROEs

Neighborhood *n* can contain the kinds of forces listed in Section 7.1.2 : NF, UF, and C.[17] Attrition occurs when two forces *a* and *b* are antagonists as defined in this section.

7.7.1 Civilian Antagonists

This model is deferred until after Spiral One.

A civilian "force" is simply a neighborhood group. Two neighborhood groups are antagonistic if at

17 We are ignoring organization group "forces" in Spiral 2A.

least one regards the other as an ENEMY. We can compute the degree of antagonism between groups *nf* and *ng* as follows:

$$antagonism_{nfg} = \min(R_{nfg}, R_{ngf})$$

If $antagonism_{nfg}$ falls in the ENEMY range, then groups *nf* and *ng* are antagonists, and will cause attrition to each other.

7.7.2 Attacking ROEs: UF

In Spiral 2A, uniformed forces (UF) may attack non-uniformed forces (NF) in a neighborhood.

A UF will attack a particular NF only when directed to do so by its Rules of Engagement. In principle, each UF has an *attacking ROE* with respect to each NF This ROE may be set to

- ATTACK
- DO_NOT_ATTACK

An ATTACK ROE is further qualified by a Cooperation limit: the UF will attack only if its composite cooperation with the residents of the neighborhood exceeds some value.

In practice, Athena will only track ROEs set to ATTACK; if no ROE is set, no attacks will take place.

For example, consider the following Attacking ROE table, in which BLUE is a uniformed force and ALQ is a non-uniformed force.

Nbhood	***f***	***g***	**Coop. Limit**
N1	BLUE	ALQ	25
N2	BLUE	ALQ	40

This table indicates that Blue has been ordered to attack Al Qaeda in neighborhood N1 provided that cooperation of the residents of N1 with Blue is at least 25%, and in neighborhood N2 provided that cooperation is at least 40%. However, Blue will not attack Al Qaeda in any other neighborhood.

7.7.3 Attacking ROEs: NF

In Spiral One, non-uniformed forces (NF) may attack uniformed forces (UF) in a neighborhood.

Just as with UFs, an NF will attack a particular UF only when ordered to do so by its Rules of Engagement. In principle, each NF has an *attacking ROE* with respect to each UF. This ROE may be set to

- HIT_AND_RUN: The NF will attempt to kill UF troops through ambushes, IED attacks, and so forth, while limiting their own losses by running as soon as possible.

- STAND_AND_FIGHT: The NF will attempt to kill UF troops by sucking them into ambushes and pinning them down, so as to kill as many UF troops as possible regardless of their own losses.

- DO_NOT_ATTACK

The HIT_AND_RUN and STAND_AND_FIGHT ROEs are further qualified by a Cooperation limit: a force will attack only if its composite cooperation with the residents of the neighborhood exceeds some value.

In addition, the HIT_AND_RUN and STAND_AND_FIGHT ROEs include a nominal number of attacks per day.[18]

In practice, Athena will not track ROEs set to DO_NOT_ATTACK; if neither HIT_AND_RUN nor STAND_AND_FIGHT is set, no attacks will take place.

For example, consider the following Attacking ROE table, in which BLUE is a uniformed force and ALQ is a non-uniformed force.

Nbhood	***f***	***g***	**ROE**	**Coop. Limit**	**Attacks/Day**
N1	ALQ	BLUE	STAND_AND_FIGHT	40	0.2
N2	ALQ	BLUE	HIT_AND_RUN	40	0.5

Al Qaeda will attack Blue in both neighborhoods, given at least 40% cooperation. In N1 it will try to attack once every five days, and will stand and fight, taking greater casualties. In N2, however, it will try to attack more often, every other day, but will try to conserve its strength by using hit and run tactics.

7.7.4 Defending ROEs

In addition to its Attacking ROEs, each uniformed force has a Defending ROE, which can have one of three values:

- FIRE_BACK_IMMEDIATELY: Fire back immediately if fired upon. This will cause collateral damage to civilians whenever the UF defends itself.

18 In a later spiral, this number will be computed automatically from the resources available to the NF.

- FIRE_BACK_IF_PRESSED: Do not fire back unless the enemy continues to engage. This will cause collateral damage to civilians only when the UF defends itself against an NF with an attacking ROE of STAND_AND_FIGHT.

- HOLD_FIRE: Do not fire back. The UF will never cause collateral damage when defending.

In Spiral One this ROE affects only the response of a UF when attacked by an NF. It is defined for every UF in every neighborhood, and defaults to FIRE_BACK_IF_PRESSED.

7.8 The Attrition Cycle

Attrition will be computed at regular intervals; the default interval is one week, i.e., 7 ticks,[19] which is comfortably smaller than the expected Athena game turn of 1 to 3 months. Thus, attrition that occurs during a game turn can have an impact during that game turn.

At the end of the attrition interval, the following algorithm will compute the attrition for the interval:

For each neighborhood *n,*
    Determine each pair *a*,*b* of antagonists in *n*.
    For each pair *a*,*b*
        Compute attrition for each pair, according to the kind of antagonists.
            UF vs. NF
            NF vs. UF
        Accumulate the attrition to each force.
    Apply all attrition at the end.

Note that the outcome is independent of the order in which the pairs are processed, as the attrition is applied to each force after all attrition has been computed. This is standard for Lanchester attrition models: all the bullets are fired, and then they all hit at once.

7.9 Computing Attrition

7.9.1 Uniformed vs. Non-uniformed

Non-uniformed forces operate in small cells and hide among the civilian population. ROE permitting, a UF will attack an NF cell every time it gets a chance, and will do so with overwhelming force. Every cell found will be destroyed, and collateral damage to civilians is likely. There will be no UF casualties in Spiral 2A as a result of UF attacks on NF cells.

The UF must find the NF cells in order to attack them. The number of cells found increases with:

- The number of troops in the UF: the more troops, the more chance of contact.

19 Model parameter: `aam.ticksPerTock`

- Increased cooperation of the civilians with the UF, because the UF will get more intel.
- Decreased cooperation of the civilians with the NF, because the civilians are less likely to warn the NF of an impending attack.
- The number of troops in the NF: the more troops, the easier they are to find.
- In a later spiral: The exposure of the UF troops to the population (i.e., the UF units are assigned activities that imply contact with the locals); you can't find the bad guys while guarding the base.

Let

ΔT	=	The duration of the attrition interval in days
Ω_{NF}	=	The composite cooperation of the civilians in *n* with the NF.
Ω_{UF}	=	The composite cooperation of the civilians in *n* with the UF.
N_p	=	The possible number of attacks by the UF on the NF during the interval.
N_a	=	The actual number of attacks by the UF on the NF during the interval
cov_{UF}	=	The actual coverage fraction of the UF .[20]
$nomcov_{UF}$	=	The nominal coverage fraction of the UF for this algorithm.[21]
cov_{NF}	=	The coverage fraction of the NF.[22]
$nomcov_{NF}$	=	The nominal coverage fraction of the NF for this algorithm.[23]
TF	=	The average time to find an NF cell, given equal cooperation and the nominal coverage fractions.[24] This is the parameter that drives this part of the model.

Then, we compute the possible number of attacks as follows:

$$N_p = \text{round}\left(\frac{\Omega_{UF}}{\max(\Omega_{NF}, 10.0)} \times \frac{cov_{UF}}{nomcov_{UF}} \times \frac{cov_{NF}}{nomcov_{NF}} \times \frac{\Delta T}{TF}\right)$$

Thus, when the cooperation levels are balanced ($\Omega_{NF} = \Omega_{UF}$), and when the coverage fractions are at their nominal levels, we get precisely the average number of attacks during the interval

20 Computed in-line using a coverage function; model parameter: `aam.UFvNF.UF.coverageFunction`
21 Model parameter: `aam.UFvNF.UF.nominalCoverage` e.g., 0.3 (depends on what would be expected as average in this scenario)
22 Computed in-line using a coverage function; model parameter: `aam.UFvNF.NF.coverageFunction`. This is identical to the default COERCION coverage function.
23 Model parameter: `aam.UFvNF.NF.nominalCoverage`, e.g., 0.4
24 Model parameter: `aam.UFvNF.UF.timeToFind`, e.g., 5 days

$$N_p = \text{round}\left(\frac{\Delta T}{TF}\right)$$

The possible number of attacks is increased by increased cooperation of the population with the UF, and by increased coverage for either group, and is decreased by increased cooperation of the population with the NF, as desired.

Next, we must determine the actual number of attacks, and the attrition resulting from them. Let

cellsize	=	The average number of troops per NF cell.[25]
$personnel_{NF}$	=	The number of NF troops in the neighborhood.

The number of cells is then

$$N_{cells} = \text{ceiling}\left(\frac{personnel_{NF}}{cellsize}\right)$$

Since each attack kills an entire cell, the actual number of attacks cannot exceed the number of cells:

$$N_a = \min(N_p, N_{cells})$$

The number of NF troops killed is then

$$N_{killed} = \min(N_a \times cellsize, personnel_{NF})$$

Next, we must compute civilian casualties. Let

$\hat{\Omega}_{UF}$	=	The nominal composite cooperation of the neighborhood with the UF for this algorithm.[26]
ECDA	=	The Expected Collateral Damage per Attack, i.e., the number of civilians killed for each attach on an NF cell, assuming nominal cooperation. This value will depend on the urbanization level of the neighborhood (rural, suburban, urban). For example, if *ECDA* is 2.0, then we expect two civilian casualties for each cell killed.[27]

Then

25 Initially, this will be a model parameter, `aam.UFvNF.NF.cellSize`, e.g., 7 NF/cell in every neighborhood. Later, it might be allowed to vary by group and by neighborhood.

26 Model parameter: `aam.UFvsNF.UF.nominalCooperation`, e.g., 35 %

27 Model parameter: `aam.UFvsNF.ECDA.`*`urbanization`*, e.g., 1 in rural, 3 in suburban, and 5 in urban.

$$N_{civcas} = \text{floor}\left(N_a \times ECDA \times \frac{\hat{\Omega}_{UF}}{\max(\Omega_{UF}, 10.0)}\right)$$

If the actual cooperation of the neighborhood with the UF is exactly the nominal, then the civilian casualties will be just as expected. If the nominal cooperation is 50%, then better than nominal cooperation can cut casualties in half—but minimal cooperation (10% or less) can increase casualties by a factor of five.

7.9.2 Non-uniformed vs. Uniformed

An NF will attack a UF every time it can (given the ROE to attack). Obviously, the NF can only attack if both the NF and the UF have personnel in the neighborhood. However, the number of such attacks is limited by the NF's resources[28], the availability of UF target opportunities, the cooperation of the civilian population with the NF, and the desire of the NF to limit their casualties as indicated by their ROE (HIT_AND_RUN or STAND_AND_FIGHT). In particular, the number of potential attacks should:

- Vary inversely with UF security
- Vary directly with the cooperation of the neighborhood with the NF.
- Vary directly with the coverage of UF units in the neighborhood.

First we compute N_p, the number of potential attacks:

$$N_p = \text{round}\left(\frac{100 - security_{UF}}{100} \times \frac{\Omega_{NF}}{\hat{\Omega}_{NF}} \times \frac{cov_{UF}}{nomcov_{UF}} \times rate \times \Delta T\right)$$

where

$rate$	=	The nominal attack rate per day.[29]
$security_{UF}$	=	The UF's security in the neighborhood. Security ranges from -100 to +100.
Ω_{NF}	=	The composite cooperation of the neighborhood with the NF.
$\hat{\Omega}_{NF}$	=	The nominal cooperation of the neighborhood with the NF for this algorithm.[30]

28 In a later spiral.

29 Ideally, the number of attacks should depend on the NF's resources, so that attacking the resources will reduce the number of attacks. Our notion is that the nominal number of attacks is determined by a Z-curve whose X-axis is a measure of the resources available to the NF. The currently envisioned model is that this measure is the weighted sum of the economic clout of the actors that have influence over the NF, weighted by the relationships between the NF and the actors. All this must wait until a later spiral, however. For now, the attack rate per day is part of the NF's attacking ROE.

30 Model parameter: `aam.NFvsUF.`*`roe`*`.nominalCooperation`, e.g., 50%.

		This parameter depends on the ROE, HIT_AND_RUN or STAND_AND_FIGHT.
cov_{UF}	=	The actual coverage fraction of the UF, based on total personnel in the neighborhood.[31]
$nomcov_{UF}$	=	The nominal coverage fraction of the UF for this algorithm.[32] This parameter depends on the ROE, HIT_AND_RUN or STAND_AND_FIGHT.
ΔT	=	The duration of the attrition interval in days.

If N_p is 0, then of course the NF cannot attack. Otherwise, whether the NF will actually attack or not depends on the ROE, HIT_AND_RUN or STAND_AND_FIGHT.

Loss Exchange Ratio

The loss exchange ratio (LER) for an attack is the number of NF casualties for each UF trooper killed. If the NF killed four UF personnel at a loss of one NF personnel, that would be an LER of ¼. AAM determines the loss exchange ratio (LER) as follows:

$ELER_{ROE}$	=	The Expected Loss Exchange Ratio: the expected number of NF casualties per UF casualty, when the UF fires back. This number depends on the NF's attacking ROE.[33]
Ω_{NF}	=	The composite cooperation of the neighborhood with the NF.
Ω_{UF}=	=	The composite cooperation of the neighborhood with the UF.
$MAXLER_{ROE}$	=	The maximum loss exchange ratio the NF is willing to accept when attacking with the specified ROE.[34]

Then

$$LER_{ROE} = ELER_{ROE} \times \frac{\Omega_{UF}}{\max(\Omega_{NF}, 10)}$$

In other words, the loss exchange ratio depends on the intelligence available to the NF and to the UF, as indicated by the cooperation of the neighborhood with each. As UF's intel improves, the LER gets larger; as NF's intel improves, the LER gets smaller. If the neighborhood cooperates equally with both, it's a wash and the LER is simply the expected LER.

The NF will only attack if the LER is their favor. That is,

---

31 Model parameter: `aam.NFvsUF.UF.coverageFunction`, e.g., {25 1000}.
32 Model parameter: `aam.NFvsUF.UF.nominalCoverage`, e.g., 0.2
33 Model parameter, `aam.NFvsUF.`*`roe`*`.ELER`, e.g., 0.33 for HIT_AND_RUN, 3.0 for STAND_AND_FIGHT.
34 Model parameter, `aam.NFvsUF.`*`roe`*`.MAXLER`, e.g., 0.25 for HIT_AND_RUN, 4.0 for STAND_AND_FIGHT.

If $LER_{ROE} \leq MAXLER_{ROE}$
Then attack,
Otherwise do not attack.

NF and UF Casualties

Hit-and-Run: When the ROE is HIT_AND_RUN, the NF will husband their forces, trying to do damage to the UF without losing too many people. We assume that the NF wants to inflict $UFCAS_{ATTACK}$ casualties on the UF during each attack.[35] Since they would not be attacking unless the Loss Exchange Ratio were in their favor, we know that they are prepared to take $NFCAS_{ATTACK}$ casualties in each attack, where

$$NFCAS_{ATTACK} = UFCAS_{ATTACK} \times LER_{ROE}$$

Stand-and-Fight: When the ROE is STAND_AND_FIGHT, on the other hand, the NF is prepared to suffer significant casualties in order kill UF personnel. We assume that the NF is willing to expend $NFCAS_{ATTACK}$ personnel to kill as many UF personnel as they can.[36] Given the LER, they can then kill

$$UFCAS_{ATTACK} = \frac{NFCAS_{ATTACK}}{\max(LER_{ROE}, 0.01)}$$

In either case, the number of NF casualties actually incurred depends on the Defending ROE of the UF: unless they fire back, no NF personnel will be killed. However, the NF must make their plans presuming that the UF will fire back.

Now, the NF can potentially make N_p attacks, given their access to the UF forces. The actual number of attacks is limited by the NF and UF personnel available. The NF cannot kill more UF personnel than are there, and will do so with the fewest casualties to themselves; and since they must presume that the UF will fire back they cannot schedule more attacks than they have personnel to lose. Thus, the actual number of attacks N_a is computed as follows.

$$N_a = \text{floor}\left[\min\left(N_p, \frac{personnel_{NF}}{NFCAS_{ATTACK}}, \frac{personnel_{UF}}{UFCAS_{ATTACK}}\right)\right]$$

Given that an attack is possible, and the LER is in the NF's favor, the NF will **always** attack at least once.

35 Model parameter, `aam.NFvsUF.HIT_AND_RUN.ufCasualties,` e.g., 4.
36 Model parameter, `aam.NFvsUF.STAND_AND_FIGHT.nfCasualties`, e.g., 20

The total number of UF casualties is then

$$UFCAS_{TOTAL} = \text{floor}\left(\min\left(N_a \times UFCAS_{ATTACK}, personnel_{UF}\right)\right)$$

NF and civilian casualties depend on whether or not the UF fires back, as shown in the following table:

NF Attacks	UF Defends	UF Fires Back
HIT_AND_RUN	FIRE_BACK_IMMEDIATELY	Yes
HIT_AND_RUN	FIRE_BACK_IF_PRESSED	No
HIT_AND_RUN	HOLD_FIRE	No
STAND_AND_FIGHT	FIRE_BACK_IMMEDIATELY	Yes
STAND_AND_FIGHT	FIRE_BACK_IF_PRESSED	Yes
STAND_AND_FIGHT	HOLD_FIRE	No

If the UF fires back, then the number of NF casualties is

$$NFCAS_{TOTAL} = \text{floor}\left(\min\left(UFCAS_{TOTAL} \times LER_{ROE}, personnel_{NF}\right)\right)$$

Otherwise, no NF casualties are incurred.

Civilian Collateral Damage

For Spiral 2A we assume no collateral damage from the NF attack itself—NF does not want to kill civilians in these attacks, but would be glad if UF fired back and did kill some. Thus, civilian casualties occur only if the UF fires back. In this case, the total number of civilian casualties is

$$CIVCAS_{total} = ECDC \times NFCAS_{TOTAL}$$

where

ECDC = the Expected Collateral Damage per NF Casualty. This is a model parameter which depends on the urbanization level (urban, suburban, or rural) of the neighborhood, and on the NF's ROE.[37]

37 Model parameter: `aam.NFvsUF.ECDC.`*`urbanization`*, e.g., 0.1 for RURAL, 0.15 for SUBURBAN, 0.2 for URBAN

7.10 Applying Attrition

All attrition is computed before any attrition is applied to the neighborhood. Attrition to a group is applied to each of a group's units in the neighborhood in proportion to its size. The attrition applied to each unit is also applied to the unit's staffing pool. Organization and civilian attrition is saved for later assessment by the DAM rule sets.

First, we build a list of the units in the neighborhood that belong to the group being attrited. (Note that civilian units from other neighborhoods are treated as belonging to distinct groups.)

Next, we compute the total number of personnel in the list of units.

Next, we compute the fraction of the total represented by each of the units.

Next, we sort the list of units in decreasing order of size.

Next we apply attrition to each unit in turn, attriting it by its proportional share of the casualties. Let

casualties	=	The total number of casualties to inflict
i	=	The index of the *i*th unit to receive casualties
f_i	=	The fraction of *casualties* to be taken by *i*.

Then,

Let r = *casualties*
For each unit i to be attrited,
  Let $k = \min\left(r, \text{ceiling}\left(f_i \times casualties\right)\right)$
  Apply k casualties to r.
  Let $r = r - k$
  If r is 0, then stop.

This algorithm rounds fractional casualties in favor of the smaller units; the smallest unit will tend to get less than its "fair" share of casualties.

7.11 Assessing the Attitude Implications

Once attrition has been computed and applied, it is necessary to assess the implications for civilian and organization attitudes via the CIVCAS and ORGCAS rule sets in the Driver Assessment Model (DAM).

7.11.1 Contrasted with JNEM

In theory, JNEM assesses civilian and organization casualties incident by incident—in theory, because it is the ground model's responsibility to decide what constitutes an incident, and some ground models do a better job than others. In consequence, JNEM accumulates all attrition to a group occurring within a short window, and calls that an "incident". From this attrition, JNEM computes the effective number of kills, and passes this through a Z-curve to get a multiplier which is used to scale the magnitudes in the CIVCAS rules.

In general, then, attrition happens when it happens, and the attitude effects of different incidents can interfere with each other (based on sharing the same cause) as they play out over time.

In Athena, we assess attrition periodically, nominally once a week, reflecting the incidents that have implicitly occurred over the previous week. So long as we approach attrition in this aggregate way, there's no way to assess it incident by incident. It is clearly wrong to say, "There were 17 fire fights this week in which Punjabis were killed in this neighborhood, so we'll have 17 inputs to GRAM all timestamped today and all with the same cause." Consequently, we will assess the attitude implications once per week as well, based on the aggregate attrition over the week.

Magic attrition to civilian and organization groups will be applied to those groups as it occurs, but the total attrition will be saved and assessed with the normal attrition at the end of the week.

7.11.2 Satisfaction Effects of Attrition

We will assess the satisfaction effects of attrition on a civilian or organization group as follows. First, all attrition occurring during the week, both magic and normal, to group *f* in neighborhood *n* will be accumulated in the **attrit_nf** table. Let

n	=	The neighborhood. For civilian units, this is the neighborhood of origin, rather than the neighborhood in which the attrition occurred. (They will often be the same, of course.)
f	=	The attrited civilian or organization group. If *f* is a civilian group, it will be resident in neighborhood *n*.
ftype	=	The group type, CIV or ORG.
casualties	=	The total number of casualties to *f* in *n* during the week.
$ZSAT_{ftype}()$	=	A Z-curve, dependent on the group type, which converts a total number of casualties into a casualty multiplier used in the CIVCAS or ORGCAS satisfaction rules.[38]
M	=	The casualty multiplier.

We compute the casualty multiplier, *M*, as follows:

38 Model parameters: `dam.CIVCAS.Zsat, dam.ORGCAS.Zsat`

$$M = ZSAT_{ftype}(casualties)$$

When the CIVCAS or ORGCAS rule set is triggered, it has access to *n*, *f*, *casualties*, and *M*.

7.11.3 Cooperation Effects of Attrition

Athena only tracks the cooperation of neighborhood groups with force groups; consequently, we assess cooperation effects only for attrition to civilian groups. The CIVCAS rule set attends to this.

All normal civilian attrition is (at present) due to collateral damage resulting from fighting between two force groups. Magic attrition can optionally be attributed to one or two force groups. As attrition occurs, Athena accumulates the total casualties to group *f* in neighborhood *n* in which force group *g* was in some way involved in the **attrit_nfg** table. If two force groups are involved in an altercation, as is usually the case, the total civilian casualties are attributed equally to both.

Then, let

n	=	The neighborhood of origin, rather than the neighborhood in which the attrition occurred. (They will often be the same, of course.)
f	=	The attrited civilian group, resident in *n*.
g	=	A force group.
casualties	=	The total number of casualties to *f* in *n* during the week in which *g* was involved.
R_{nfg}	=	The relationship between neighborhood group *nf* and force group *g*.
$ZCOOP()$	=	A Z-curve which converts a total number of casualties into a casualty multiplier used in the CIVCAS coop rules.[39] It will usually be the same as $ZSAT_{CIV}()$
M	=	The resulting casualty multiplier.

We compute the casualty multiplier, *M*, as follows:

$$M = ZCOOP(casualties)$$

When the CIVCAS rule set is triggered it has access to *n*, *f*, *g*, *casualties*, R_{nfg}, and *M*. The actual magnitude of the rule firing will be

$$magnitude = M \times \text{enmore}\left(R_{nfg}\right) \times \text{M-}$$

where M- connotes a medium-sized negative (i.e., -5.0 point) effect and "enmore" is the "enemies more" relationship multiplier function. See the *Athena Rules* document for details on how rule magnitudes are specified.

39 Model parameter: `dam.CIVCAS.Zcoop`

8. DEMOGRAPHICS

The Athena Demographics model (DEMOG) models the number of people in the civilian population broken down in a variety of ways. In the long run, this model will handle births, deaths due to old age and other causes, and aging of the population; in Spiral 2A, it will track the following:

- The base population, by neighborhood group broken down into two groups: those who support themselves by means of subsistence agriculture and barter (the subsistence population), and those who participate in the regional economy (also known as consumers).
- The number of deaths due to attrition for each neighborhood group.
- Resident population: people in their neighborhood of origin.
- Displaced population: people assigned activities outside of their neighborhood of origin.[40]

And, taking all of these into account,

- The total number of consumers in each neighborhood.
- The total labor force present in a neighborhood, including both resident and displaced population present in the neighborhood. In computing the labor force, we assume that the subsistence population is not included in the job market.

8.1 Requirements for Spiral 2A

The Spiral 2A demographics model is intended to be as simple as possible while meeting the needs of the other Athena models. In particular:

- GRAM requires the current population of each neighborhood group.
- The Ground model requires that the civilian population of a neighborhood can be assigned activities in other neighborhoods.
- The Athena Attrition Model (AAM) requires that the civilian population can take collateral damage as the result of combat between forces.
- The Economics Model requires the number of people who participate in the regional economy (the consumers) and the number of people in the potential labor force (the workers).

40 This term is somewhat ambiguous. Civilians can be assigned the DISPLACED activity while remaining in their own neighborhood; in this case, they are considered "resident" rather than "displaced".

Consequently,

- The playbox population is initially stated as the population of each neighborhood group.
- Each neighborhood group has a percentage of its population (possibly zero) that supports itself by subsistence agriculture and therefore does not participate in the regional economy. This is the subsistence population. The remainder of the population are the consumers.[41]
- Resident population can be *displaced*; that is, it can be assigned activities outside of its neighborhood of origin.
  - The reason for the displacement can be indicated by the choice of activity, e.g., to DISPLACED or IN_CAMPS; however, this is outside the scope of DEMOG.
- The labor force is a fraction of the consumers in the neighborhood, disregarding those displaced personnel who are not in a position to work. (I.e., those in camps, or new in the neighborhood).

8.2 Simplifying Assumptions

In order to keep DEMOG within the realm of the possible we make the following simplifying assumptions in Spiral 2A:

- Births, and deaths from causes other than attrition, are not tracked.
- The population does not age.
- The subsistence population is a simple percentage of the non-displaced population of each neighborhood group.
- The subsistence population is outside the regional cash economy.
- Displaced personnel are displaced from their land, and clearly cannot be doing subsistence agriculture.[42]
- The labor force is a simple fraction of the total consumers, taking civilian activities (e.g., refugee status) into account.

41 In reality, of course, the dividing line between subsistence and consumption will not be so stark; subsistence farmers will often buy or trade for certain goods, and some consumers may also grow crops or animals for their own use. As a first approximation, however, this suffices.

42 Nomadic herdsmen are clearly a separate case, and one that must wait until we implement migration fully. Consequently, any subsistence herdsman and assumed to stay within their neighborhoods of origin (for now).

8.3 Population and Units

Athena 1 had the distinction between explicit population (represented in units) and implicit population (the default). In Athena 2, units are an output rather than an input: the ground model "staffs" units based on scheduled activities for the purpose of visualization and to ease subsequent computations. As a result, every civilian is represented both implicitly and explicitly, and consequently the distinction is no longer of interest.

8.4 Neighborhood Group Population

The resident population of neighborhood group *ng* at the current simulation time is

$$population_{ng} = BP_{ng} - DP_{ng} - attrition_{ng}$$

where

BP_{ng}	=	The base population of group *ng* at time zero, as defined in the scenario.
DP_{ng}	=	The displaced population: the total number of personnel belonging to group *ng* but assigned activities in some other neighborhood.
$attrition_{ng}$	=	The total number of casualties suffered by group *ng* to date.

8.4.1 Civilian Attrition

In Athena V2, all attrition occurs to personnel in units created by the ground model's staffing algorithm, but the actual attrition is applied to the pool of people from which the units are staffed.[43] For civilians, consequently, attrition to a neighborhood group's population is handled by adding the casualties to $attrition_{ng}$:

$$attrition_{ng} = attrition_{ng} + casualties$$

8.4.2 Subsistence Population

The subsistence population of a neighborhood group, SP_{ng}, is a fraction of the total resident population:

$$SP_{ng} = \left(\frac{SAP_{ng}}{100.0}\right) * population_{ng}$$

43 This has the unfortunate side effect that if DISPLACED personnel belonging to group *ng* are attrited, they will be re-staffed from their neighborhood of origin. However, the use of the DISPLACED activity to represent civilians displaced from their homes is a stopgap measure, pending enhancements to GRAM that will allow a portion of a neighborhood group to truly move to another neighborhood. Consequently, we'll live with the problem for now.

where

SAP_{ng}	=	The Subsistence Agriculture Percentage for neighborhood group *ng*.

Note that displaced personnel are *never* part of the subsistence population. This opens a slight hole in the model: if the displaced personnel are brought back to their neighborhood of origin, some of its personnel might once again be counted as part of the subsistence population. This is unrealistic, as people who leave subsistence agriculture are rarely able to go back, as they generally have lost their land and livestock.

8.4.3 Consumer Population

The consumer population of a neighborhood group is simply that part of the resident population that participates in the regional economy, i.e., that are not members of the subsistence population. Thus, we define neighborhood group *ng*'s consumer population CP_{ng} as follows:

$$CP_{ng} = population_{ng} - SP_{ng}$$

We will account for consumers in the group's displaced population at the neighborhood level, rather than as part of a neighborhood group.

8.4.4 Labor Force

Civilians contribute to the labor force to different degrees, depending on their assigned activities, and in addition, we must exclude the subsistence population. Since the entire civilian population is now represented in units, we define the contribution to the labor force[44] of group *ng* as follows:

$$LF_{ng} = \left(\frac{100.0 - SAP_{ng}}{100.0}\right) \times \left[\sum_{\substack{u \in ng \\ u \notin \text{displaced}}} LFF(a_u) \cdot personnel_u\right]$$

where

SAP_{ng}	=	The subsistence agriculture percentage for group *ng*.
u	=	A unit belonging to group *ng* and located in neighborhood *n*.
a_u	=	The activity assigned to unit *u*
$personnel_u$	=	The number of personnel in unit *u*
$LFF(a_u)$	=	The labor force fraction for activity *a*.

Note that we will account for displaced workers at the neighborhood level, rather than as part of a neighborhood group.

44 By "labor force" we mean that portion of the population that seeks to be employed, whether they are in fact employed or not.

The Labor Force Fractions are model parameters[45], changeable at run-time; the default values are as follows:

Unit Activity	**Labor Force Fraction**
NONE	0.6
DISPLACED	0.4
IN_CAMP	0.0

The assumption is that DISPLACED units, which are mingled with the host population, look for jobs at a decreased rate due to their unsettled condition. As DISPLACED units assimilate, they can be given an activity of NONE. Further, we assume that personnel in refugee camps have no opportunity to look for work. Again, if camps turn into permanent settlements the unit activity can be changed.

8.5 Neighborhood Population

Neighborhood figures are the total of the neighborhood group figures, plus the total of the figures for units displaced from their neighborhoods of origin. Thus, we must first determine the figures for displaced units.

8.5.1 Displaced Personnel

The total personnel in units displaced from their neighborhood of origin is simply

$$DP_n = \sum_{\substack{u \in n \\ u \in \text{displaced}}} personnel_u$$

where

u	=	A unit in neighborhood n, but displaced from its neighborhood of origin.
$personnel_u$	=	The number of personnel in unit u.

8.5.2 Displaced Consumers

To be displaced is to be displaced from one's land; hence, all displaced personnel must willy-nilly participate in the regional economy. Thus, all displaced personnel are consumers.

45 `demog.laborForceFraction.`*`activity`*

8.5.3 Displaced Labor Force

The displaced labor force in neighborhood *n*, DLF_n, is computed much like the labor force for a neighborhood group:

$$DLF_n = \sum_{\substack{u \in n \\ u \in \text{displaced}}} LFF(a_u) \times personnel_u$$

8.5.4 Neighborhood totals

The total population of neighborhood *n* is simply the population of all resident neighborhood groups, plus all displaced personnel:

$$population_n = DP_n + \sum_{g \in \text{CIV}} population_{ng}$$

The consumer population and labor force are computed similarly:

$$CP_n = DP_n + \sum_{g \in \text{CIV}} CP_{ng}$$

$$LF_n = DLF_n + \sum_{g \in \text{CIV}} LF_{ng}$$

8.6 Regional Population

The regional population, consumers, and labor force are simply summed up across the neighborhoods in the obvious way.

8.7 Unemployment

Unemployment can drive attitude change. The Economics Model computes the unemployment rate, *UR*, for the region of interest.

8.7.1 Disaggregation to Neighborhoods

Lacking any better way to disaggregate unemployment, we will assume that *RealUR* affects each neighborhood in proportion to its labor force. That is, if *RealUR* is 5%, then 5% of each neighborhood's labor force will be unemployed. The unemployed population for each neighborhood group is then UP_n:

$$UP_n = LF_n \times \frac{UR}{100.0}$$

where

LF_n = The number of people in the labor force in neighborhood *n*.

However, the size of the labor force relative to the neighborhood population as a whole depends on the subsistence population of each neighborhood group. A high unemployment rate may be of little concern in a neighborhood with 90% subsistence. For attitude effects, we are primarily concerned with the ratio of unemployed workers with the total population: the unemployed workers per capita, or UPC_n:

$$UPC_n = 100.0 \times \frac{UP_n}{population_n}$$

8.7.2 Disaggregation to Neighborhood Groups

Precisely the same logic applies to each group in each neighborhood. The unemployed population for neighborhood group *ng* is UP_{ng}:

$$UP_{ng} = LF_{ng} \times \frac{UR}{100.0}$$

where

LF_{ng} = The number of people from group *ng* in the labor force in neighborhood *n*.

And then, the unemployed per capita for each group *ng* is:

$$UPC_{ng} = 100 \times \frac{UP_{ng}}{population_{ng}}$$

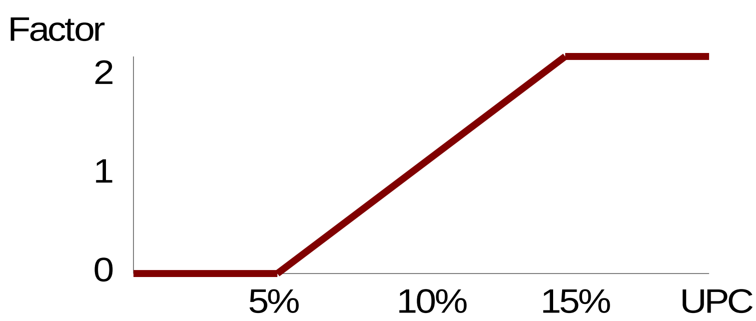
8.7.3 Unemployment Situations

How does unemployment affect the civilians?

- The unemployed civilians take a hit to their satisfaction levels, with the usual indirect effects on other groups. This will particularly affect Quality of Life (QOL), and should depend on each neighborhood group's effective unemployment rate.

- If there are many unemployed civilians, the fact of their presence will become a problem for their neighbors due to increases in crime and other social conflict. This effect will be negligible for small unemployment rates, and dominant for large ones (for some value of "small" and "large"). This will particularly affect Safety (SFT), and possibly Autonomy (AUT), and should depend on the effective unemployment rate of the neighborhood as a whole.

If unemployment in a neighborhood is sufficiently dire, it should engender an unemployment situation to drive an UNEMPLOYMENT rule set. This is a new kind of situation, a "demographic situation" rather than an environmental or activity situation. If we follow the pattern used for activity situations, then we will want to compute some kind of factor, analogous to a coverage fraction, that will be used to scale the magnitudes in the situation's rule set. We will use a Z-curve to convert the unemployment per capita into a multiplicative factor, ranging from 0.0 to 2.0:



The above curve says that the unemployment rules will not begin to affect general attitudes until about 5% of the general population are unemployed; the rules reach their nominal magnitude when 10% of the population are unemployed; and the rules have their maximum effect when 15% are unemployed. This is not to say that problems do not continue to get worse when more than 15% of the population are unemployed; but very high unemployment brings other problems that can't simply be addressed by an attitude rule set.

Using the above curve, we will compute an unemployment atttidue factor (UAF) for the neighborhood as a whole, and for each individual group within it:

$$UAF_n = Z_{UAF}(UPC_n)$$

And

$$UAF_{ng} = Z_{UAF}(UPC_{ng})$$

An unemployment situation will exist in for neighborhood group *ng* if $UAF_n > 0.0$, or if $UAF_{ng} > 0.0$ for any *ng*. The situation will be assessed by a new UNEMP rule set

The modeling team proposed the following strawmen for the rule set:

- QOL: $UAF_{ng} \times \text{L}-$

- SFT: $UAF_n \times \text{M}-$
- AUT: $UAF_n \times \text{S}-$

We assume that a neighborhood group's QOL is affected primarily by unemployment within the group, but that SFT and AUT are primarily affected by the number of unemployed workers in the neighborhood as a whole. For example, one would expect the crime rate to rise with the number of unemployed workers.

The UNEMP situation will have a near factor of p = 0.2, and a far factor of q = 0.0.

9. ECONOMICS

The Athena Economics model (ECON) 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 will refer to this as the *local economy*. The core of ECON is a 3-sector Computable General Equilibrium (CGE) model solved as a system of non-linear equations using the Gauss-Seidel algorithm. A complete narrative description of the derivation and content of the 3-sector CGE is pending; this document will give a brief overview, and then explain how the CGE is embedded in ECON and how it relates to the rest of Athena. The CGE itself is implemented as a "cell model"; the cell model code is included in Section 9.13 .

9.1 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 region. The unit of production is the *goods basket* (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 in the local economy that isn't covered by the other 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* (EBasket), which is similar to the goods basket and also nominally costs about $1.

9.2 The Economic Tableau

ECON displays the current state of the economy in a spreadsheet-like tableau:

	goods	**pop**	**else**	**Revenue**	**Price**	**Quantity**	**Units**
goods	X_{gg}	X_{gp}	X_{ge}	REV_g	P_g	QD_g	GBasket/yr
pop	X_{pg}	X_{pp}	X_{pe}	REV_p	P_p	QD_p	work-year/yr
else	X_{eg}	X_{ep}	X_{ee}	REV_e	P_e	QD_e	EBasket/yr
Expense	EXP_g	EXP_p	EXP_e				

where

X_{ij}	=	The payment in \$/year from sector *j* to sector *i*.
EXP_j	=	The total expenditure, in \$/year, of sector *j*.
REV_i	=	The total revenue, in \$/year, of sector *j*.
P_i	=	The price, in \$, of one unit of production of sector *i*.
QD_i	=	The quantity of *i*'s product demanded per year.

In short, the CGE determines how much each sector produces, what each unit of production costs, the revenue and expenditure for each sector, and (from these) the Consumer Price Index (CPI), the Gross Domestic Product (GDP), and the Deflated Gross Domestic Product (DGDP).

The following equations are true, either by definition or whenever the economy is in equilibrium:

$$EXP_j = \sum_i X_{ij}$$

$$REV_i = \sum_j X_{ij}$$

$$REV_i = P_i \times QD_i$$

$$REV_i = EXP_i$$

9.3 Text Notation

The CGE itself is implemented as a *cell model*, a pseudo-spreadsheet in which every cell has a name rather than a row and column. The cell model is contained in a text file, and so we also have a plain text notation for each of the mathematical symbols given above:

X_{ij}	=	`X.i.j`
EXP_j	=	`EXP.j`
REV_i	=	`REV.i`
P_i	=	`P.i`
QD_i	=	`QD.i`

When we are referring to a particular sector or sectors, the sector names are spelled out in full in the plain text form:

X_{gp} = `X.goods.pop`

9.4 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 needs increase in size. Decrease the amount of consumption, and the economy must needs 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 is driven by the number of consumers; and the number of consumers is given to the Economic model by the Demographic model. The shape of the economy is determined when the economy is calibrated; see Section 9.6 .

9.5 Production Functions

The shape of the economy is largely determined by the technologies used by the sectors to produce their products; and these technologies are described by *production functions*.

A production function determines how much product a sector consumes (from its own and other sectors) to produce one unit of its own product. All three sectors are modeled using the Cobb-Douglas production function, which implies that, when ingredient quantities are chosen to minimize costs, the sector will spend a fixed proportion of its money on each of the three sectors, i.e., 0.5 on **goods**, 0.4 on **pop**, and 0.1 on **else**.[46] As prices change, the Cobb-Douglas production function allows the sectors to trade off the product of one sector for the product of another.

These fixed proportions are called the Cobb-Douglas parameters, and are denoted f_{ij}.

9.6 Calibrating the CGE

The CGE is calibrated by setting a number of model parameters:

- `econ.f.`*`i`*`.`*`j`*, the Cobb-Douglas parameters for the three sectors.
- `econ.BaseWage, the average wage for one work-year, in dollars.`
- `econ.GBasketPerCapita`, the average consumption of **goods** by each consumer per year, in goods baskets.

In theory we should also set the base prices for the **goods** and **else** sectors; but as the unit of production for these sectors is an arbitrarily-sized basket worth nominally $1, the base price for these sectors is naturally $1.

Athena has preset values for these parameters. If desired, the analyst can choose other values based on the economy of the region of interest, as described in the following subsections. These parameter values should be set during Scenario Preparation, before the scenario is locked and enters time 0.

46 Another choice is the Leontief production function, which says that each unit produced requires a fixed amount of the products of the other sectors, e.g., 3 cups of flour, 2 eggs, and so forth. The Leontief production function is a better match for certain sectors in real economies, and it is likely that we will use it in the future.

9.6.1 Fill in the Social Accounting Matrix

The Social Accounting Matrix, or SAM, is the upper left portion of the economic tableau:

	goods	**pop**	**else**	**Revenue**
goods	BX_{gg}	BX_{gp}	BX_{ge}	$BREV_g$
pop	BX_{pg}	BX_{pp}	BX_{pe}	$BREV_p$
else	BX_{eg}	BX_{ep}	BX_{ee}	$BREV_e$
Expense	$BEXP_g$	$BEXP_p$	$BEXP_e$	

The analyst must determine, from whatever sources, the base flow of money to each sector from each sector (the BX_{ij}'s), in dollars,[47] and compute the base revenues and expenses: the row and column sums, $BREV_i$ and $BEXP_j$. In a typical SAM, the revenues will equal the expenses.

9.6.2 Compute the Cobb-Douglas Parameters

Given the base SAM and the Cobb-Douglas production assumptions, the analyst can compute the Cobb-Douglas parameters. Let

$$f_{ij} = \frac{BX_{ij}}{BEXP_j}$$

Note that the f_{ij}'s sum to 1.0 down each column; hence, of the nine values we only need to enter six model parameters:

- `econ.f.goods.goods`
- `econ.f.goods.pop`
- `econ.f.goods.else`
- `econ.f.pop.goods`
- `econ.f.pop.pop`
- `econ.f.pop.else`

47 Athena's concept of dollars is somewhat notional, and certainly isn't tied to the buying power of a real American dollar. The analyst can actually use any monetary units he chooses, provided he is consistent.

9.6.3 Set the Base Wage and Consumption

Next we need to provide the link between shape of the economy and its size. We do this by setting the base wage and the consumption rate.

The base wage is the average wage in dollars for one work-year of work, where work-year is defined as in Section 9.1 . The model parameter is `econ.BaseWage`. This determines each worker's purchasing power.

Then, we set the consumption of goods baskets (GBasket) per capita per year: how much **goods** each consumer consumes on average. This figure will usually be rather less than the average wage, because each basket initially costs one dollar, consumers consume **pop** and **else** as well as **goods**, and not all consumers are workers. The model parameter is `econ.GBasketPerCapita`.

9.7 Scenario Inputs

The following input values are plugged into the CGE by the ECON model as part of the base scenario:

Cell	Source	Description
`BaseConsumers`	DEMOG	The total number of consumers in the local region at time 0. This number is used to size the economy, and calibrates a number of constants used subsequently.

9.8 Run-time Inputs

The following input values are plugged into the CGE by the ECON model at each "tock", that is, at each update of the CGE as time passes:

Cell	**Source**	**Description**
`In::Consumers`	DEMOG	The number of consumers in the local region at the current. This number drives consumption, which determines the size of the unconstrained economy.
`In::WF`	DEMOG	The number of workers in the local economy at the current time. This number determines the production constraint for the **pop** sector, and also drives the computation of unemployment.
`In::LSF`	ECON, Ground	The Labor Security Factor, a number from 0.0 to 1.0. The LSF decreases with neighborhood security; (1-LSF) is the the fraction of the work force that stays home from work out of fear for their lives. Thus, this number also affects the

Cell	Source	Description
		production constraint for the **pop** sector.
`In::CAP.goods`	Ground	Each neighborhood can produce a certain quantity of goods; this quantity is calibrated at time 0 based on the size of the labor force in that neighborhood. The maximum possible production is the capacity of the economy to produce goods, which constrains the size of the economy. The contribution of each neighborhood to CAP.goods can be increased or decreased using the neighborhood's Production Capacity Factor (PCF).

9.9 Outputs

The CGE produces the following ouput values.

Cell	Used By	Description
`Out::P.`*`i`*	Display	The price of one unit of sector *i*, in dollars.
`Out::QS.`*`i`*	Display	The quantity supplied for sector i, i.e., the number of units produced.
`Out::REV.`*`i`*	Display	The revenue of sector *i*, that is, `Out::P`*`i`*`*Out::QS.`*`i`* in dollars.
`Out::EXP.`*`i`*	Display	The expense of sector *i*, that is, the dollars spent on the ingredients for the product of sector *i*.
`Out::QD.`*`i.j`*	Display	The quantity of sector *j*'s output purchased by sector *i*.
`Out::X.`*`i.j`*	Display	The dollars spent by sector *i* on sector *j*'s output.
`Out::LATENTDEMAND.`*`i`*	Display	The additional quantity of sector *i*'s product that the economy would purchase if it could be produced. (**goods** and **pop** only)
`Out::IDLECAP.`*`i`*	Display	The additional quantity of sector *i*'s product that the sector could produce if only there were demand for it. (**goods** and **pop** only)
`Out::Unemployment`	Display	The number of workers who are currently unemployed, including normal turbulence.
`Out::UR`	DEMOG	The unemployment rate, as a percentage.
`Out::GDP`	Display	The Gross Domestic Product, in dollars: the total revenue of the economy, excluding the **else** sector.
`Out::CPI`	Display	The Consumer Price Index, which measures changes in

Cell	Used By	Description
		buying power since the start of the simulation.
`Out::DGDP`	Display	The Deflated Gross Domestic Product, i.e., the GDP divided by the CPI. This is the current size of the economy, in "time 0" dollars.

9.10 Ways to Affect the Economy

The Economy is affected at each economic tock by the inputs listed in Section 9.8 . Consequently, the following things taking place in Athena as a whole will affect the economy:

- Civilian casualties can decrease the number of consumers and workers.
- Subsistence population, when displaced from their land, willy-nilly become consumers; they might not be able to contribute to the work force, depending on their assigned activity.
- When a civilian group's security in a neighborhood decreases, workers stay home out of fear, thus reducing the effective size of the work force. This is measured by the Labor Security Factor (LSF).
- Each neighborhood's Production Capacity Factor can be increased or decreased, reflecting building of new plant or destruction of existing plant with the consequent effect on CAP.goods.

9.11 Ways the Economy Affects Athena

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. This is done in the Demographic model. See Section 8.

9.12 CGE Architecture

The CGE equations are implemented as a *cell model*[48]; the cell model is given in Section 9.13 . A cell model is like a non-GUI spreadsheet model, in which each every cell has a name rather than row and column indices. Cells can contain constants or formulas; the application can plug in input values by setting the values of constant cells, and can read outputs by retrieving the value of formula cells. The model is broken up into a number of pages, each with a well-defined purpose:

- The **null** page contains calibration constants, and values computed from them, that are used in later pages. The values on the **null** page never change after the CGE is calibrated.

48 See the cellmodel(5) man page for a complete description of cell model syntax and semantics.

- The **Cal** page contains a system of non-linear equations which are iterated to a solution using the Gauss-Seidel algorithm when the CGE is calibrated. During this step a number of values are computed which are used as the basis for the evolution of the CGE over time.

- The **In** page contains all inputs to the CGE that vary as the time progresses. Athena updates these values prior to every economic tock.

- The **U** page defines the unconstrained model of the economy over time. It inherits most of its equations from **Cal**, with the necessary changes to use **Cal**'s outputs and **In**'s inputs. It computes the economy as it would be if there were no labor or goods capacity constraints.

- The **C** page inherits the equations from **U**, and modifies them to apply labor and goods capacity constraints. Starting with the outputs from **U**, and based on the Cobb-Douglas assumptions, it adjusts the prices and quantities for **goods** and **else** such that the constraints are not exceeded.

- The **Out** page contains all outputs used or displayed by Athena. In addition to copying many of **C**'s outputs directly, it also computes the unemployment rate, `Out::UR`, the CPI, `Out::CPI`, and the Deflated Gross Domestic Product, `Out::DGDP`.

9.13 CGE Content

```
# -*-Tcl-*-
#-----------------------------------------------------------------------
# TITLE:
#    eco3x3.cm, version s
#
# AUTHOR:
#    Bob Chamberlain
#    Will Duquette
#
# DESCRIPTION:
#    Prototype CGE for the Athena Economics Model.  This is a
#    3x3 Cobb-Douglas model, based on Ian Sue Wing's MIT paper.
#
#    Criteria for success:
#
#    * The model converges.
#    * All REV.i = EXP.i
#    * Prices and quantities are reasonable
#      * Prices should recover the values used to calibrate the
#        constants A.i.
#    * Quantities produced should equal the sum of the demands.
#      * I.e., deltaQ.i should = 0 for all i.
#    * Quantities should not all be zero.
#
# PAGES:
#    The model contains the following pages.  We expect the whole
#    model to be computed at time 0, as part of calibration; as time
#    advances, the model will be recomputed periodically starting at
```

```
#    page In.
#
#    "null" Basic inputs, including the Base Case and possibly SAM
#           data, and non-iterative calibration.
#    Cal    Iterative calibration based on data from the null page.
#    In     Application-settable inputs for the U and C pages.
#    U      Unconstrained model; the size of the economy is driven by
#           consumer demand.
#    C      Constrained model; the size of the economy is constrained
#           by the production capacity and the labor supply, both of
#           which are supplied by the rest of Athena.
#    Out    Output page: computes overages, shortages, and idle capacity
#           (e.g., unemployment) by comparing the constrained and
#           unconstrained results, as well as other outputs to the
#           rest of Athena.
#
#    NOTE: The Athena application is aware only of the "null", In, and Out
#    pages.
#
# HISTORY:
#    Version s: Workers who stay home from work due to low security
#               decrease CAP.pop but are not considered to be
#               unemployed.  Also, added labor turbulence to
#               to LATENTDEMAND.pop.
#
#    Version r: Revised the formulas for C::QS.goods and C::QS.pop so that
#               in the absence of binding constraints they will be at least
#               as big as on the U page.
#
#               Added a number of diagnostic cells to the Out page, to
#               support the application's sanity check of the results.
#
#    Version q: Added CSF and LSF, so that low security can decrease
#               both consumption and the labor supply.
#               Added ediff() to the deltaQ.i and deltaREV.i formulas,
#               so that it's easier to see whether they are met or not.
#               Copied the values of deltaQ.i and deltaREV.i to the
#               Out page, in case we want the application to keep
#               an eye on them.  Note that CSF has no affect at this time.
#
#               Added a new command to initialize the C page from the
#               U page each time C is solved.  The C page does not
#               stand alone; it is explicitly an adjustment of the
#               unconstrained result.
#
#    Version p: Corrected computation of CPI on page C to be based on
#               consumer purchases rather than all purchases.  Removed
#               CAP.else once again, as it is no longer needed.
#               Renamed the names BasePopulation and In::population
#               with the correct names BaseConsumers and
#               In::Consumers.  Added computation of GDP.
#
#    Version o: Added capacity limits for the else sector to facilitate
#               further testing.  Deleted the "deltaP" parameters, as
#               they aren't needed.  Other cosmetic changes.
```

```
#
#    Version n: Corrected the price adjustment equations so they
#               correctly use the optimal Cobb-Douglas demands to
#               adjust prices so that demand equals supply (i.e.,
#               markets clear).  This is justified by the assumption
#               that there are no economies of scale in any sector.
#               Changed numeraire for the price equations on the C
#               page to U::P.pop, the average wage in the
#               unconstrained case.  Revised QD.goods.pop to maintain
#               f.goods.pop under reduced production.
#
#    Version m: The price adjustments in version l affect the CPI, so
#               the P.goods implied by the base case CPI, U::P.goods,
#               is used as the numeraire instead of the CPI.
#
#    Version l: Adjusts prices when production constraints are encountered
#               on page C until demand reduces enough to achieve market
#               clearance.  Based on version j, not on version k.
#
#    Version k: Attempted to achieve market clearance when production is
#               constrained by simply limiting QD.i.j.  Didn't work;
#               some QD.i.j went negative during iteration.
#               Constraining QD.i.j >= 0.0 still didn't work.  Attempt
#               abandoned for further analysis.
#
#    Version j: Replaced [Cal::SUM] in U::P.goods with [SUM]; [SUM]
#               is now computed as part of pages U and C.  In addition,
#               SHORTAGE.i is really the latent demand for the product
#               of section i, so it is renamed LATENTDEMAND.i.  OVERAGE.i
#               turns out not to be useful, so it has been removed.
#
#    Version i: Added computation of labor market turbulence as a
#               percentage of the labor force.  Added explicit
#               computation of the non-turbulent, "real",
#               unemployment.  Made minor cosmetic improvements.
#
#    Version h: Re-integration of version g into Athena.
#               * Added Out::BQS.goods = Cal::QS.goods, to support the
#                 CAP.goods calibration.
#
#    Version g: Re-ordered the equations per RGC.
#
#    Version f: Restored the notion that the work force should be
#               endogenous by calibrating the per capita demands on
#               the null page, then assuming jobs will be driven by the
#               demand for labor, but limited by the possibly changing
#               workforce statistics (population, available work force).
#
#    Version e: Made the CPI the numeraire.  Assumed the demographics
#               model will compute both CAP.pop and QS.pop, i.e., the
#               demographics model owns the unemployment rate.
#
#               WHD: It's reasonable that Demographics should compute
#               the size of the workforce given wages and other
#               opportunities.  But the CGE must compute the number
```

```
#                of people who actually *can* work given production
#                constraints.  (This assumes that idle goods production
#                capacity, if any, can be put to work as demand rises
#                in the equilibrium economy.)
#
#                WHD: In this version, the cal page and the U page could
#                be merged; however, I'm going to leave them be.
#
#    Version d: Revised P.pop and A.pop: per RGC, they are defined
#                just like the other P.i's and A.i's.  Also, completed
#                distinction between Quantity Supplied (QS.i) and
#                Quantity Demanded (QD.i, QD.i.j).
#
#    Version c: Added QS.i, Quantity Supplied, with REV.i = P.i*QS.i.
#
#    Version b: Set In::CAP.goods and In::CAP.pop to 1e15 initially;
#                they are set by econ(sim) and should be effectively
#                infinite until then.
#
#                Copied X.i.j, Q.i.j, P.i, Q.i, REV.i, EXP.i from C
#                to out, to make them visible to econ(sim).
#
#    Version a: Based on prototype cd3x3r.cm.
#                Added In:: page, distinguished between base case inputs
#                and dynamic inputs (e.g., BasePopulation and
#                In::population).
#
#-----------------------------------------------------------------------

#-----------------------------------------------------------------------
# Indices

index i     {goods pop else}
index ing   {pop else}
index j     {goods pop else}
index imost {goods pop}

#=======================================================================
# Null Page
#
# The "null" page contains cells that are global to the rest of the
# model, and that do not change as simulation time advances.  Some are
# inputs to the model; others are computed from the inputs and should
# not be changed.  In principle, many of these parameters are "calibrated"
# from a Social Accounting Matrix (SAM).

# Normal turbulence in employment

let TurFrac = 0.04 ;# Average fraction of workers "temporarily" unemployed

# SAM-based Parameters
#
# A Social-Accounting Matrix can in principle be used to calibrate the
# model, as described here.  HOWEVER, from Athena's point of view the
# SAM data should be used only to determine the "shape" of the economy,
```

```
# i.e., the f.i.j's; the "size" of the economy must be driven
# by the Ground and Demographic models.
#
# If there is SAM data, it is:
#
#   BX.i.j     The payment in $/year from sector j to sector i
#   BP.j       The price of one unit of the product of sector j
#
# Then compute:
#
#   BREV.i      = sum.j BX.i.j
#   BEXP.j      = sum.i BX.i.j
#   BQD.i.j     = BX.i.j/BP.i
#   f.i.j       = BX.i.j/BREV.j
#   A.goods.pop = BQD.goods.pop/BaseConsumers
#
# If there is no SAM, we need to input BaseConsumers, A.goods.pop,
# f.i.j, and BP.j.
#
# Since the SAM specifies both "size" and "shape", whereas the
# f.i.j's specify only "shape", and since we must determine the "size"
# from the Ground and Demographic models, we prefer to input the
# f.i.j's rather than the BX.i.j's.  The f.i.j's can, of course,
# be computed from a SAM ahead of time.

#-----------------------------------------------------------------------
# Scenario Inputs

let BaseConsumers = 1e6     ;# Number of consumers in the inital population.

#  f.i.j is the fraction of j's revenue that is spent in sector i.

let f.goods.goods = 0.2
let f.pop.goods   = 0.4
let f.else.goods  = {1 - [f.goods.goods] - [f.pop.goods]}

let f.goods.pop   = 0.75
let f.pop.pop     = 0.1
let f.else.pop    = {1 - [f.goods.pop] - [f.pop.pop]}

let f.goods.else  = 0.3
let f.pop.else    = 0.05
let f.else.else   = {1 - [f.goods.else] - [f.pop.else]}

# Base prices: These are the sector prices used to calibrate the
# Cobb-Douglas coefficients.

let BP.goods = 1       ;# $/goodsBKT
let BP.pop   = 400     ;# $/work-year
let BP.else  = 1       ;# $/elseBKT

# Base quantities demanded by pop.  The population spends its
# income, BREV.pop, on the sectors according to the f.i.pop's.
# We begin with the per-capita consumption of goods, A.goods.pop,
# and the base number of consumers.  From this, we compute BQD.goods.pop.
```

```
# Given the price of goods and the fraction that the population
# spends on goods, we get BREV.pop.  We then use the f.i.pop's and
# P.j's to compute the other BQD.i.pop's.
#
# The BQD.i.pop's are used in the equation for P.goods.  Ultimately,
# the "size" of the economy depends on A.goods.pop and the population.

let A.goods.pop    =  114    ;# Direct consumption of goods, in
                              # goodsBKT/year per capita

let BQD.goods.pop  =  {
    [BaseConsumers]*[A.goods.pop]
} -value 1.14e8    ;# goodsBKT/year

let BREV.pop       =  {
    [BQD.goods.pop]*[BP.goods]/[f.goods.pop]
} -value 1.52e8

let BQD.pop.pop    =  {
    [f.pop.pop]*[BREV.pop]/[BP.pop]
} -value 38000     ;# work-year/year

let BQD.else.pop   =  {
    [f.else.pop]*[BREV.pop]/[BP.else]
} -value 2.28e7    ;# elseBKT/year

#-----------------------------------------------------------------------
# Calibration Constants
#
# These values are computed from the inputs above, and should not
# be modified.

# Cobb-Douglas production function coefficients.  The following
# formulas compute the calibrated Cobb-Douglas coefficients A.i that
# should yield the P.i = BP.i when the model is solved.
#
# The A.j's describe the technology via the production function
# (and the utility via the utility function).  Athena might want to
# change these assumptions eventually, especially for different kinds
# of sectors.

let A.pop = {
    <:prod i {([BP.$i]/[f.$i.pop])**[f.$i.pop]}:> / [BP.pop]
} -value 0.0094501

let A.else = {
    <:prod i {([BP.$i]/[f.$i.else])**[f.$i.else]}:> / [BP.else]
} -value 2.975941843

# Base CPI: defines the CPI for the base case; indexes the CPI to the
# start of the simulation.
let BCPI = 1.0

#========================================================================
# Calibration
```

```
#
# The following page is used to calibrate the CGE during scenario
# preparation, based on data from the null page.  It is not recomputed
# as time advances.

#-----------------------------------------------------------------------
# Calibration Page
#
# This page defines the basic CGE equations, and solves by using data
# from the null page, sizing the economy by population's demand for
# goods in the base case.

page Cal

# REV.i is the income of sector i: the product of P.i * QS.i, where
# QS.i is the quantity supplied of i's product.
#
# EXP.j is the expenditures by sector j on the various
# sectors i: sum of the X.i.j's down the column.  At present, it is
# used only for output, to verify that EXP.j=REV.j.

# REV.i = P.i * QS.i
# EXP.j = sum.i X.i.j

define REV.i {i} {[P.$i]*[QS.$i]}
define EXP.j {j} {<:sum i {[X.$i.$j]}:>}

let REV.goods = [REV.i goods]                        -value 2.92125e8
let EXP.goods = [EXP.j goods]                        -value 2.92125e8

let REV.pop   = [REV.i pop]                          -value 1.52e8
let EXP.pop   = [EXP.j pop]                          -value 1.52e8

let REV.else  = [REV.i else]                         -value 3.99e8
let EXP.else  = [EXP.j else]                         -value 3.99e8

# X.i.j is the revenue sector i receives from sector j; it's computed
# as i's share of j's total revenue.
#
# X.i.j = f.i.j * REV.j

define X.i.j  {i j} { [f.$i.$j] * [REV.$j] }

let X.goods.goods  = {<:X.i.j goods goods:>}  -value 5.8425e7
let X.pop.goods    = {<:X.i.j pop goods:>}    -value 1.1685e8
let X.else.goods   = {<:X.i.j else goods:>}   -value 1.1685e8
let X.goods.pop    = {<:X.i.j goods pop:>}    -value 1.14e8
let X.pop.pop      = {<:X.i.j pop pop:>}      -value 1.52e7
let X.else.pop     = {<:X.i.j else pop:>}     -value 2.28e7
let X.goods.else   = {<:X.i.j goods else:>}   -value 1.197e8
let X.pop.else     = {<:X.i.j pop else:>}     -value 1.995e7
let X.else.else    = {<:X.i.j else else:>}    -value 2.5935e8


# QD.i.j is number of i's units "purchased" by j at price P.i;
```

```
# it's simply the dollar amount divided by the price.
#
# QD.i.j = X.i.j / P.i
#
# Note that QD.goods.pop is special, as it drives the size of the
# economy.

define QD.i.j {i j} { [X.$i.$j] / [P.$i] }

let QD.goods.goods = {<:QD.i.j goods goods:>} -value 5.8425e7
let QD.pop.goods   = {<:QD.i.j pop goods:>}   -value 292125
let QD.else.goods  = {<:QD.i.j else goods:>}  -value 1.1685e8
let QD.pop.pop     = {<:QD.i.j pop pop:>}     -value 38000
let QD.else.pop    = {<:QD.i.j else pop:>}    -value 2.28e7
let QD.goods.else  = {<:QD.i.j goods else:>}  -value 1.197e8
let QD.pop.else    = {<:QD.i.j pop else:>}    -value 49875
let QD.else.else   = {<:QD.i.j else else:>}   -value 2.5935e8

# Some sort of exogenous demand is required to size the
# economy, so we have chosen to size the economy (in the unconstrained
# case) based on the per capita consumer demand for goods.

# NOTE: We'll redefine this in U as A.goods.pop*In::consumers
let QD.goods.pop = {[BQD.goods.pop]} -value 1.14e8

# QD.i is the demand for the product of sector i in the sector's units.
# It is computed as the sum of the sector-by-sector demands for sector
# i's product.
#
# QD.i = SUM.j(QD.i.j)

define QD.i {i} {
    <:sum j {[QD.$i.$j]}:>
}

let QD.goods = {<:QD.i goods:>} -value 2.92125e8
let QD.pop   = {<:QD.i pop:>}   -value 380000
let QD.else  = {<:QD.i else:>}  -value 3.99e8

# In the unconstrained case, the quantities supplied, QS.i, are made
# equal to the quantity demanded, QD.i.  This is Walras' Law that
# supply = demand at equilibrium.

let QS.goods = {[QD.goods]}     -value 2.92125e8
let QS.pop   = {[QD.pop]}       -value 380000
let QS.else  = {[QD.else]}      -value 3.99e8

# When j is a production sector, the price of one unit of its product,
# P.j, is obtained by inserting the expressions for the optimal values
# of the ingredients,
#
#   QD.i.j = f.i.j*P.j*QS.j/P.i,
#
# into the condition that REV.j=EXP.j, where REV.j is the product of
# the price, P.j, and the quantity supplied, QS.j, and EXP.j is computed
```

```
# by summing the expenses.  We use the Cobb-Douglas production function,
#
#   QS.j = A.j * PROD.i (QD.i.j ** f.i.j)
#
# to express the supplied quantity in terms of the quantities of
# ingredients, and insert the expressions for those quantities.  The
# QS.j drop out of the equation and we solve for P.j.  The solution is
#
#   P.j = (PROD.i (P.i/f.i.j)**f.i.j)/A.j.
#
# Goods and else are production sectors; pop may be treated like one,
# with consumption "producing" the labor.

define P.j {j} {
    <:prod i {([P.$i]/[f.$i.$j])**[f.$i.$j]}:> / [A.$j]
}

# let P.goods = [P.j goods] -value 1.0
let P.pop   = [P.j pop]   -value 400.0
let P.else  = [P.j else]  -value 1.0

# However, the price equations are homogeneous, so one of them is
# useless, and the above equation for P.goods is not used. Instead,
# we use the BCPI (base case consumer price index) as the numeraire that
# defines the value of the $ in terms of a weighted sum of prices,
# where the weights are consumption by consumers in a base case.
# Solving BCPI=1 for P.goods gives:

let C       = {<:sum i {[BP.$i] * [BQD.$i.pop]}:>}      -value 1.52e8
let SUM     = {<:sum ing {[P.$ing] * [BQD.$ing.pop]}:>} -value 3.8e7
let P.goods = {([C]*[BCPI] - [SUM])/[BQD.goods.pop]}    -value 1

# P.goods was chosen because QD.goods.pop, which becomes a
# divisor, is never zero.

#-----------------------------------------------------------------------
# Diagnostics

# deltaQD.i.pop verifies that QD.i.pop = BQD.i.pop at the end of
# calibration.

define deltaQD.i.pop {i} {
    ([QD.$i.pop] - [BQD.$i.pop])/
    max(1.0, [QD.$i.pop], [BQD.$i.pop])
}

let deltaQD.goods.pop = {<:deltaQD.i.pop goods:>}
let deltaQD.pop.pop   = {<:deltaQD.i.pop pop:>}
let deltaQD.else.pop  = {<:deltaQD.i.pop else:>}

# deltaQ.i: Verifies that QS.i = SUM.j QD.i.j.  The value of deltaQ.i
# should be within an epsilon of 0.0.

define deltaQ.i {i} {
    ediff(0.0, 1.0 - <:sum j {[QD.$i.$j]}:>/[QS.$i])
```

```
}

let deltaQ.goods = {<:deltaQ.i goods:>}
let deltaQ.pop   = {<:deltaQ.i pop:>}
let deltaQ.else  = {<:deltaQ.i else:>}

# deltaREV.i: Verifies that REV.i = SUM.j X.i.j.  The value of deltaREV.i
# would be within an epsilon of 0.0.

define deltaREV.i {i} {
    ediff(0.0, ([REV.$i] - <:sum j {[X.$i.$j]}:>)/[REV.$i])
}

let deltaREV.goods = {<:deltaREV.i goods:>}
let deltaREV.pop   = {<:deltaREV.i pop:>}
let deltaREV.else  = {<:deltaREV.i else:>}

#========================================================================
# Dynamic Pages
#
# The following pages are recomputed at each "tock".  Inputs from the
# rest of Athena and from the user that can change as time passes
# appear on the In page.  Outputs to Athena appear on the Out
# page.

#------------------------------------------------------------------------
# Inputs page
#
# The values given on this page are notional; the real values will
# come from outside the CGE.

page In

# Current Consumer Price Inflator
let Inflator   = 1.0

# Consumption Security Factor: decreases consumption due to low
# neighborhood group security.
let CSF        = 1.0

# Labor Security Factor: decreases labor due to low neighborhood group
# security.
let LSF        = 1.0

# Number of consumers currently in the population.
let Consumers = 1e6          ;# people.

# Max capacity for each sector.  These are set by Athena; the initial
# values are intended to be effectively infinite.

# Max production rate for goods
let CAP.goods  = 1e15         ;# goodsBKT/year

# Work Force: Number of people who want to be employed
let WF         = 400000       ;# work-years/year
```

```
# Due to the normal turbulence, some of those in the Work Force
# are temporarily unemployed and hence not available to work. Thus,
# CAP.pop is the WF less this turbulence.  In addition, low security
# can reduce the effective workforce because people are afraid to
# go to work.  In a later version, WorkersAtHome can include strikers,
# who also limit capacity but are not considered to be unemployed.

let Turbulence    = {[WF]*[TurFrac]}
let WorkersAtHome = {([WF] - [Turbulence])*(1.0 - [LSF])}
let CAP.pop       = {[WF] - [Turbulence] - [WorkersAtHome]}

# NOTE: Because else includes the rest of the world, it is presumed to
# have unlimited capacity even in the constrained case.  Hence, there
# is no CAP.else.

# let CAP.else = 1e15


#------------------------------------------------------------------------
# Unconstrained Page
#
# This pages runs the CGE for the unconstrained solution, using
# equations copied from the Cal page, sizing the economy based on
# demand for goods by the current population.

page U

copypage Cal -except {
    C
    deltaQD.goods.pop
    deltaQD.pop.pop
    deltaQD.else.pop
}

# QD.goods.pop is now computed from the current population, not
# from the base population.
let QD.goods.pop = {[A.goods.pop]*[In::Consumers]} -value 1.14e8

# P.goods is now based on the current Consumer Price Index, rather
# than the Base CPI.
let P.goods = {([Cal::C]*[In::Inflator] - [SUM])/[BQD.goods.pop]} -value 1

let CPI = {
    <:sum i {[P.$i]  * [BQD.$i.pop]}:> / <:sum i {[BP.$i] * [BQD.$i.pop]}:>
} -value 1.0

let deltaCPI = {
    ([CPI] - [In::Inflator])/[In::Inflator]
}


#------------------------------------------------------------------------
# Constrained Page
#
```

```
# This page runs the CGE for the constrained solution, i.e., it takes
# the current labor force and production capacity into account and may
# therefore impose limits on the Quantity Supplied (QS.i) by the
# goods and pop sectors, as compared with the unconstrained solution.
# When demand exceeds capacity, prices increase to force market clearance.
#
# Otherwise, the model is the same as on the U page.

page C
copypage U
initfrom U

let QD.goods.pop = {
    [f.goods.pop]*[REV.pop]/[P.goods]
} -value 1.14e8

# The quantity supplied is constrained by the production capacity; if
# the constraint is not binding, it should be at least as big as on
# the U page.
let QS.goods = {
    min([In::CAP.goods], max([QD.goods], [U::QD.goods]))
} -value 2.92125e8

let QS.pop = {
    min([In::CAP.pop], max([QD.pop], [U::QD.pop]))
} -value 380000

# The capacity of else is unconstrained.  This entry doesn't really
# change anything, but it makes the assumption explicit.
let QS.else = {[QD.else]}   -value 3.99e8

let P.goods = {
    ([f.goods.pop]*[REV.pop] + [f.goods.else]*[REV.else]) /
    ((1.0 - [f.goods.goods])*[QS.goods])
} -value 1

# The value of P.pop implied by CPI=1 in the unconstrained case is the
# numeraire for the constrained case.  Note that P.pop is already
# inflated by the In::Inflator.

let P.pop = {[U::P.pop]} -value 400

let P.else = {
    ([f.else.goods]*[REV.goods] + [f.else.pop]*[REV.pop]) /
    ((1.0 - [f.else.else])*[QS.else])
} -value 1


#========================================================================
# Page Out: The outputs.

page Out

# Copy base case outputs from Cal
let BQS.goods = {[Cal::QS.goods]}
```

```
# Copy outputs from C
foreach i {goods pop else} {
    let P.$i        = {[C::P.$i]}
    let QS.$i       = {[C::QS.$i]}
    let REV.$i      = {[C::REV.$i]}
    let EXP.$i      = {[C::EXP.$i]}
    let deltaQ.$i   = {[C::deltaQ.$i]}
    let deltaREV.$i = {[C::deltaREV.$i]}

    foreach j {goods pop else} {
        let QD.$i.$j = {[C::QD.$i.$j]}
        let X.$i.$j  = {[C::X.$i.$j]}
    }
}

let CPI = {[C::CPI]}

# goods shortages and overages: goodsBKT/year
let LATENTDEMAND.goods = {max(0.0, ediff([U::QD.goods],   [C::QS.goods]))}
let IDLECAP.goods      = {max(0,0, ediff([In::CAP.goods], [C::QS.goods]))}

# Unemployment Statistics
let RealUnemployment = {
    max(0.0, ediff([In::CAP.pop] + [In::WorkersAtHome],[C::QS.pop]))
}  ;# People

let Unemployment     = {[RealUnemployment] + [In::Turbulence]} ;# People

# pop shortages and overages: work-years/year
let LATENTDEMAND.pop   = {
    max(0.0, ediff([U::QD.pop],   [C::QS.pop])) + [In::Turbulence]
}
let IDLECAP.pop        = {max(0,0, ediff([In::CAP.pop], [C::QS.pop]))}

# Unemployment Rates, real and reported
let RealUR = {100.0 * [RealUnemployment]/[In::WF]} ;# Percentage
let UR     = {100.0 * [Unemployment]/[In::WF]}     ;# Percentage

# Gross Domestic Product, GDP

let GDP  = {
    [X.goods.pop] + [X.pop.pop] + [X.goods.else] + [X.pop.else]
}

# GDP deflated by CPI, which is used as a proxy for the GDP deflator.
let DGDP = {[GDP]/[CPI]}

# Sanity Check Values
let SUM.QS = {
    ediff([QS.goods] + [QS.pop] + [QS.else], 0.0)
}

let FLAG.QS.NONNEG = {
    [QS.goods] >= 0.0 || [QS.pop] >= 0.0 || [QS.else] >= 0.0
```

```
}

let FLAG.P.POS = {
    [P.goods] > 0.0 || [P.pop] > 0.0 || [P.else] > 0.0
}

let FLAG.DELTAQ.ZERO = {
    [deltaQ.goods] == 0.0 && [deltaQ.pop] == 0.0 && [deltaQ.else] == 0.0
}
```

10. ACRONYMS

AAM	Athena Attrition Model
AUT	Autonomy (concern)
CIV	Civilian
CMO	Civil/Military Operations
CTR	Contractor
CUL	Culture (concern)
DAM	Driver Assessment Model
ECDA	Expected Collateral Damage per Attack
ECDC	Expected Collateral Damager per NF Casualty
ELER	Expected Loss Exchange Ratio
FRC	Force
GRAM	Generalized Regional Attitude Model
IED	Improvised Explosive Device
IGO	International or Inter-Governmental Organization
JNEM	Joint Non-lethal Effects Model
LER	Loss Exchange Ratio
LFF	Labor Force Fraction
MAD	Magic Attitude Driver
MAG	Mars Analyst's Guide
NF	Non-uniformed Force
NGO	Non-Governmental Organization
ORG	Organization
QOL	Quality of Life (concern)
ROE	Rules Of Engagement
S&RO	Stability & Recovery Operations
SFT	Safety (concern)
UF	Uniformed Force
UNESCO	United Nations Educational, Scientific, and Cultural Organization