Mars Analyst's Guide

Mars Simulation Infrastructure Library, V1.65

February, 2012

William H. Duquette

`William.H.Duquette@jpl.nasa.gov`

Jet Propulsion Laboratory

Models

1. INTRODUCTION

This document presents the models and related constructs implemented by version 1.65 of the Mars Simulation Infrastructure Library (Mars). The models are described in sufficient detail to allow implementation; the implementation itself is not in the scope of this document.

1.1 Other Mars Documents

The Mars documentation set may be found in the "mars/docs" directory of the Mars build tree; open "mars/docs/index.html" in a web browser, and follow the links. The documentation is usually included in the documentation set for client simulations. Otherwise, documents can be obtained directly from the JNEM or Athena projects; contact `David.R.Hanks@jpl.nasa.gov` or `William.H.Duquette@jpl.nasa.gov`. Note that this document is also available in hardcopy.

Software Manual Pages

Extensive documentation of the Mars software tools and libraries is included in the software installation set in the form of software "man pages".

2. MARS CONCEPTS

This section gives an overview of Mars and the concepts shared by its various models. The discussion is kept to a high level; see Sections 3 and following for detailed models.

2.1 The Client Simulation

Mars is an infrastructure library; its models are intended for use in other simulation applications. These are referred to as *client simulations*, or simply as *clients*.

2.2 MAM: Modeling Belief Systems

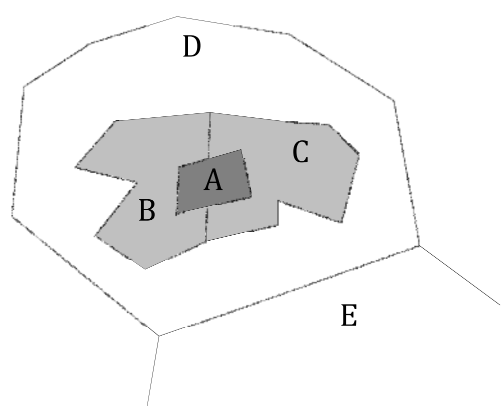
MAM, the Mars Affinity Model, models the *belief systems* of entities, e.g., *civilian groups*, and from them computes *affinities* which are the basis for *relationships*. MAM is described in detail in Section 3.

2.3 GRAM: Modeling the Population

GRAM, the Generalized Regional Attitude Model, models the attitudes of the civilian groups that reside in a particular region, and of the organization groups that work in the region. GRAM is described in detail in Section 4.

2.4 The Playbox

GRAM models population dynamics in a geographical region called the *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, and neighborhood proper. Geographically, neighborhoods are usually defined as polygons whose vertices are defined by map coordinates; however, this is the province of the client simulation. 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.



Events in the client simulation can have attitude effects in GRAM; these effects take place within neighborhoods, and affect the population of the neighborhoods. The geographic *spread* of the ripple effects of an event taking place in a neighborhood depends on how nearby other neighborhoods are presumed to be--not simply geographically but also socially. The nearness of one neighborhood to another is called the *neighborhood proximity*. There are four proximity levels: *here*, *near*, *far*, and *remote*. The diagram above shows proximity to neighborhood A. From A's point of view, A is here. Suburbs B and C are *near* A, and outlying area D is *far* from A. Neighborhood E is *remote*.[1] 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.

2.5 Groups

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

2.5.1 Civilian Groups

Civilian groups represent the population of the playbox, i.e., the people who actually live in the neighborhoods. This population maybe broken into groups by ethnicity, religion, language, social class, political affiliation, or any other demographic criteria the analyst deems necessary. 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. Civilian groups are usually united by their belief systems.

Each civilian group resides in a neighborhood, and each neighborhood must contain at least one civilian group.

GRAM models civilian groups in detail, tracking the attitudes of each group along several axes as the group's members are affected by events and situations taking place in the client simulation. These attitudes then will typically affect the group's reactions and responses in the client simulation.

Note: In earlier versions of GRAM, a single civilian group might reside in multiple neighborhoods; the group in a particular neighborhood was referred to as a neighborhood group.

---

[1] Note that these proximities are social, not geographic—proximities are input to GRAM, and are not computed from the geometry of the neighborhoods or the distance from one neighborhood to another.

2.5.2 Force Groups

Force groups represent military forces, such as the U.S. Army, and other groups whose purpose is to apply force. GRAM models the level of *cooperation* (i.e., information sharing) of civilian groups with force groups.

2.5.3 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. Organizations may be either local or foreign.

Earlier versions of GRAM tracked attitudes for organization groups. This is no longer the case; however, because all existing client simulations model organization groups, the group type is still defined in Mars.

2.6 Simulated Time

Mars measures simulated time in integer *ticks*. The duration of one tick can be anything from one second to two minutes to three hours to four or more days; tick sizes of one minute and of one day are typical. The simclock(n) module tracks simulated time, and converts between ticks and hours, minutes, and seconds; it also supports military “Zulu-time” strings.

In addition, GRAM expresses rates of attitudinal change as points per decimal day; it automatically converts between decimal days and ticks.

Client simulations will often have a minor time step, the tick, and also one or more major time steps; these are accordingly called *tocks*. In JNEM, for example, the tick is one minute, and GRAM is advanced at the tock, once every five ticks.

3. MARS AFFINITY MODEL

Athena and JNEM rely heavily on the concept of the horizontal relationship R_{fg} between two groups f and g, where $-1.0 \leq R_{fg} \leq +1.0$. Athena 3 adds the vertical relationship V_{ga} between a group g and an actor a, where $-1.0 \leq V_{ga} \leq +1.0$. Because relationships are pair-wise, a large scenario can have thousands or tens of thousands of them. The analyst can enter all of these values, but this is slow, tedious, and error-prone, even presuming that the analyst can determine what all of the relationships should be.

It seems that the nature of a group's relationships should be due to something about the groups involved. Positive relationships are due to shared cultures, values, and aims; negative relationships are due to opposed cultures, values, and aims. It seems reasonable, then, that if we could characterize the significant cultures, values, and aims in a region, and then rate each entity with respect to each of them, that we could use that as a basis to compute some notion of relationship between the two entities.

Given two entities f and g, then, we define the *affinity* of entity f for entity g to be the natural degree of relationship between the two entities given their particular *belief systems*. Affinity is denoted A_{fg}, where $-1.0 \leq A_{fg} \leq +1.0$. The relationship used by a client simulation might then be exactly equal to this affinity, or might be a function of it.

This section describes a model of belief systems and a method of computing affinities from them. I will speak primarily in terms of the affinity of one group for another; however, the discussion applies equally well to affinities between groups and actors.

3.1 Belief Systems Defined

We must characterize the values, aims, cultures, and beliefs of each group, so that we can compare them.

3.1.1 Topics

First, we define a set of *topics* numbered from 1 to *N*. Each topic represents some issue, value, cultural belief, aim, etc., that is significant in the region of interest. Topics must be stated in absolute terms so that they can be compared across groups. The statement "My party should control the government" is a relative statement, for example; group f's opinion on this topic will mean something different than group g's. The statement "Party X should control the government" is an absolute one, and something about which f and g can have opinions that can be meaningfully compared.

3.1.2 Positions

Then, let P_{fi} be the *position* of group *f* with respect to topic *i*, where $-1.0 \leq P_{fi} \leq +1.0$. A position indicates where the group stands on the topic. A position of 1.0 indicates strong support; a position of –1.0 indicates strong opposition.

The magnitude of a position,

$$Z_{fi} = |P_{fi}|$$

is called the *strength* or *zeal* of the position, and indicates the extent to which group *f* will take action in the public sphere in support of their position. It does *not* indicate group *f*'s certainty or firmness of belief. For example, it might be that group *f* despises vanilla ice cream but is sent into transports of ecstasy by chocolate ice cream. Unless these preferences lead to significant action in the public sphere, however, group *f*'s position on both flavors should be 0.0.

For groups, we can interpret P_{fi} as the position of the group as an aggregate, without making any assumptions as to how homogenous or heterogeneous the positions of the members of the group are, or whether any particular individual's position matches that of the group as a whole.

For actors, perception is more important than reality. An actor's professed beliefs may differ from his privately held beliefs; and it may be further modified by his actions, if they are inconsistent with his professed beliefs.

We will frequently use the following scale when dealing with position values[2]:

Symbol	Name	Value
P+	Passionately For	0.9
S+	Strongly For	0.6
W+	Weakly For	0.3
A	Ambivalent	0.0
W–	Weakly Against	–0.3
S–	Strongly Against	–0.6
P–	Passionately Against	–0.9

[2] This scale is implemented by the `qposition` type in simtypes(n).

3.1.3 Emphasis

The positions P_{fi} capture group *f*'s beliefs about particular topics; but to see how group *f* feels about group *g* based on their beliefs, we need to know how group *f* responds to disagreement. We define E_{fi} to be group *f*'s *emphasis* on agreement or disagreement with respect to topic *i*, where $0.0 \leq E_{fi} \leq 1.0$.

If E_{fi} is near 1.0, then group *f* puts its emphasis on agreement for topic *i*; agreement on the topic will drive affinity up and disagreement will be discounted. If E_{fi} is near 0.0, then group *f* puts its emphasis on disagreement; disagreement will drive affinity down, and agreement will be discounted.

The original concept used the term "tolerance for disagreement" rather than "emphasis". In the process of defining affinity we tried several different ways of making E_{fi} affect the computed affinities before finding a formula with reasonable behavior; and in the process of explaining the model to others we found that "tolerance" is a loaded word and did not convey the meaning we intended. The current term was chosen to be both neutral and descriptive of the parameter's role in the computation.

We will frequently use the following scale when dealing with emphasis values[3]:

Symbol	Name	Value
ASTRONG	Agreement—Strong	0.90
AWEAK	Agreement	0.70
NEITHER	Neither	0.50
DWEAK	Disagreement	0.35
DSTRONG	Disagreement—Strong	0.25
DEXTREME	Disagreement—Extreme	0.15

3.1.4 Belief Systems

The values of P_{fi} and E_{fi} for all *i* are said to constitute group *f*'s belief system.

[3] This scale is implemented by the `qemphasis` type in simtypes(n).

3.2 Modeling Affinity

3.2.1 Desired Properties

Our formula for computing A_{fg} should have the following properties:

- $-1.0 \leq A_{fg} \leq +1.0$
- Relationships of –1.0 are pathological; A_{fg} should only be –1.0 for pathological inputs.
- Affinities should be asymmetric, because different things are important to different groups. Group *f*'s affinity for g depends on the topics that are most important to *f*, while g's affinity for *f* is based on the topics that are most important to g. Therefore, A_{fg} need not equal A_{gf}, and usually won't.
- If the group *f*'s position on topic *i* differs in sign from group g's position, that's a stronger disagreement than if they do not differ in sign, even if the absolute magnitude of the difference is the same. Consider +0.5 and +0.3 vs. +0.1 and –0.1. In the first case the groups differ in the degree of their support; in the latter, one group supports and the other does not. The absolute difference is 0.2 in either case.
- Strong disagreement should not yield indifference. That is, if groups *f* and g disagree strongly on one or more topics, A_{fg} should not be 0.0.
- Zealots should distrust the lukewarm. If group *f*'s position on *i* is strong and group g agrees only weakly with *f* about *i,* then topic *i* should reduce the affinity between *f* and g.
- The lukewarm might admire the zealots. The strength of the zealot's zeal might increase the affinity of a less zealous group for them.
- The ambivalent should distrust the zealots. Even if group *f*'s position on a topic is 0.0, the group might still be nervous about those who feel strongly about the topic one way or the other, and thereby have a lower affinity with such groups than they otherwise would.[4]
- A_{fg} should be generally continuous for small changes in any input.

---

[4] We had originally thought that if P_{fi} were zero for some some topic *i,* then that topic should have no effect on A_{fg}. Later we realized that group *f* would still care about the groups around them, even if they did not care about *i.*

- The value of A_{fg} should be reasonable even when the number of topics is 1.

3.2.2 Definitions

Given N topics numbered $i = 1, \dots, N$, and two groups f and g, let P_{fi} be group f's position on topic i, and let E_{fi} be group f's emphasis on agreement vs. disagreement, as defined in Section 3.1. Then, we make the following additional definitions.

First, we define the sign and strength of the positions:

$$B_{fi} = \text{sign}(P_{fi})$$
$$Z_{fi} = |P_{fi}|$$

Group f's affinity for group g will depend on the extent to which they agree or disagree about each topic. Thus, we next define agreement and disagreement metrics for groups f and g and all topics i. The agreement metric, G_{fgi}, is defined as follows:

$$G_{fgi} = \begin{cases} \sqrt{P_{fi} \cdot P_{gi}} & \text{if } B_{fi} = B_{gi} \\ 0 & \text{if } B_{fi} \neq B_{gi} \end{cases}$$

If f and g's positions differ in sign, then they do not agree; their agreement is 0.0. Otherwise, we compute their agreement as the geometric mean of their positions. This has several useful properties:

- $0 \leq G_{fgi} \leq 1.0$
- G_{fgi} has the same "units" as Z_{fi}.
- If P_{fi} or P_{gi} is 0.0, then there is no agreement; $G_{fgi} = 0$.
- If $P_{fi} = P_{gi}$, then $G_{fgi} = |P_{fi}|$.
- Agreement is symmetrical. (Affinity will not be.)

In other words, if two groups are both lukewarm about a topic, their agreement will be lukewarm; if the two groups are passionate about the topic, their agreement will also be passionate. If one is passionate and one is lukewarm, their agreement will be somewhere in the middle. And if they do not agree at all, their agreement is 0.0.

The geometric mean is much better than the arithmetic mean in this case. Suppose that P_{fi} is 1.0, but P_{gi} is 0.0. The arithmetic mean would give an agreement of 0.5, which is absurd given that g is completely ambivalent about the topic. The geometric mean gives an agreement of 0.0 in this case.

Next, the disagreement metric, D_{fgi}, is defined as the average difference in their positions:

$$D_{fgi} = \frac{\left|P_{fi} - P_{gi}\right|}{2}$$

Again, this has several useful properties:

- $0 \leq D_{fgi} \leq 1.0$
- D_{fgi} has the same "units" as G_{fgi}; thus, agreements and disagreements are comparable.
- Disagreement is symmetrical. (Affinity will not be.)

Next, we define the emphasis ratio:

$$\beta_{fi} = \frac{1 - E_{fi}}{E_{fi}}$$

This value is used to magnify the importance of disagreements. If E_{fi} is 1.0 (a perfect emphasis on agreement) then β_{fi} is 0.0 and disagreements on topic *i* don't matter to *f*. As E_{fi} approaches zero, β_{fi} gets larger and larger until, in the pathological case of $E_{fi} = 0.0$, disagreement on topic *i* will dominate the result.

3.2.3 The Basic Model

Intuitively, the affinity between groups *f* and *g* is increased by agreements and decreased by disagreements, as modified by the emphasis that *f* puts on agreements and disagreements. If we were to compute the affinity between the two groups using only one topic *i*, we could define it like this:

$$A_{fgi} = \frac{G_{fgi} - \beta_{fi} \cdot D_{fgi}}{1 + \beta_{fi} \cdot D_{fgi}}$$

The denominator normalizes the result so that $-1.0 \leq A_{fgi} \leq +1.0$.

The affinity of *f* with *g* given all topics should clearly be some kind of weighted average of the A_{fgi}'s. Now Z_{fi}, the strength of *f*'s position on *i*, is precisely how important topic *i* is to *f* relative to other topics; hence, Z_{fi} is a natural weight. Normalizing, this gives us the following equation:

$$A_{fg} = \frac{\sum_i Z_{fi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\sum_i Z_{fi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

This equation has several special cases that would result in division by zero, and which therefore require special handling:

- As mentioned above, β_{fi} is undefined when E_{fi} is 0.0; there are several distinct cases.
- The denominator will be 0.0 if Z_{fi} is 0.0 for all *i*.

We will address the special cases in the final version of the model.

3.2.4 Handling Ambivalence

One of our requirements is that the ambivalent will distrust the zealous, that is, if Z_{fi} is 0.0 and Z_{gi} is high, then group *f*'s affinity for g will tend to decrease. At present it does not, because we use the Z_{fi}'s as the weights in our weighted average, and topics for which Z_{fi} is zero drop out.

We used the Z_{fi}'s as the weights because we wished to weigh the agreements and disagreements by the importance of the topics to group *f*. What we now see is that Z_{fi} doesn't adequately capture that importance. In fact, a topic can be important because of the strength of *f*'s belief, or because of the strength of the disagreement.

Hence, we define M_{fgi} to be the importance of topic *i* to group *f* in the context of group g's belief system, where

$$M_{fgi} = \max(Z_{fi}, D_{fgi})$$

In short, strong disagreements trump weak zeal. The equation for A_{fg} then becomes

$$A_{fg} = \frac{\sum_i M_{fgi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\sum_i M_{fgi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

or, equivalently,

$$A_{fg} = \frac{\sum_i \max(Z_{fi}, D_{fgi}) \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\sum_i \max(Z_{fi}, D_{fgi}) \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

This new formula has all of the desired properties of its predecessor, and also allows the ambivalent to distrust the zealous.

3.2.5 Implicit Commonality

While working with the two versions of the model described above, we noticed two problems:

- Affinity numbers were lower than we expected, and were frequently negative.

- The computed affinity was very sensitive to the number of topics.

We determined that the problem was with our selection of topics. When defining topics for a region, it is natural to focus on the topics for which there is disagreement; we were defining many such topics, but were omitting topics on which most of the entities agree. In short, we were neglecting the *cultural commonality* of the groups in the playbox.

3.2.5.1 Adding Implicit Topics

We realized that commonality could be handled by adding some number of implicit topics on which there is general agreement. Thus, we assume that there are some common topics not included in the list of specific topics. Let us assume further, and without loss of generality, that the two groups are in complete agreement on these topics, and are extremely passionate. That is, they have the same positions, and their zeal is 1.0.

If these topics were added to the affinity calculation, they would insert an additive term in the numerator and an identical term in the denominator. This leads to the following redefinition of A_{fg}:

$$A_{fg} = \frac{\eta_{fg} + \sum_i M_{fgi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\eta_{fg} + \sum_i M_{fgi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

where η_{fg} is the number of implicit topics of agreement between f and g. If η_{fg} is 0, we have the same model we had before. As η_{fg} increases, so does the implicit commonality of the two groups. When $\eta_{fg} = N$, implicit commonality balances the explicit topics; when $\eta_{fg} < N$, the explicit topics dominate, and when $\eta_{fg} > N$, the implicit commonality dominates.

3.2.5.2 Playbox Commonality

It is clear that η_{fg} cannot be an input to the model; it is a pairwise value, and one of the reasons for defining the affinity model is to avoid such pairwise inputs. Consequently, let us define $\eta_{playbox}$ as the cultural commonality within the playbox, and η_f as group f's share in that common culture, where $0 \leq \eta_f \leq \eta_{playbox}$. If group f belongs to the dominant culture in the playbox, then $\eta_f = \eta_{playbox}$. If group f is a complete outsider, then $\eta_f = 0$. Then we can define η_{fg} as follows:

$$\eta_{fg} = \min(\eta_f, \eta_g)$$

If f and g are both members of the dominant culture, then they will get all of the playbox commonality; if either is an outsider, they will get none of it.

However, η_f doesn't make a good input, either; ideally it ought to depend on $\eta_{playbox}$. So let

$$\eta_f = \theta_f \cdot \eta_{playbox}, \text{ for } 0 \le \theta_f \le 1$$

Set θ_f to 1 for members of the dominant culture, and to 0 for complete outsiders. We can then define

$$\eta_{fg} = \min(\theta_f, \theta_g) \cdot \eta_{playbox}$$

3.2.5.3 The Playbox Commonality Dial

Our formulation of η_{fg} does a good job of adding implicit commonality to the model. However, there is the related problem that A_{fg} is very sensitive to the number of topics. We can handle this by tying η_{fg} to the number of explicit topics as follows:

$$\eta_{playbox} = \gamma \cdot N \text{ for } \gamma \ge 0$$

The input γ is then our commonality dial. It can default to 1.0, where implicit topics exactly balance explicit topics, and be adjusted up and down by the user. Increasing or decreasing γ will tend to increase or decrease affinities across the board, to the extent that the groups involved participate in the playbox commonality.

γ has another useful property—the analyst can set the playbox commonality and then add or delete topics without needing to adjust it, because γ retains its meaning as the number of topics changes.

3.2.5.4 Summary

Our model of affinity with implicit causality taken into account can be stated as follows:

Let

γ	=	Playbox commonality dial, where $\gamma \ge 0$; the value is nominally 1.0.
θ_f	=	Entity commonality dial, where $0 \le \theta_f \le 1$; nominally 1.0 for local entities and 0.0 for non-local entities.
N	=	The number of explicit topics

Then

$$\eta_{playbox} = \gamma \cdot N$$

$$\eta_{fg} = \min(\theta_f, \theta_g) \cdot \eta_{playbox}$$

$$A_{fg} = \frac{\eta_{fg} + \sum_i M_{fgi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\eta_{fg} + \sum_i M_{fgi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

This is the fundamental affinity equation. It cannot simply be used as is, however, because there are several pathological sets of inputs; see Section 3.3

3.2.6 Are the Properties Met?

This model meets our desired properties.

- $-1.0 \leq A_{fg} \leq +1.0$.
- Affinities of –1.0 should be pathological. In fact, extreme inputs are required to reach either extreme, which is a good thing. In particular, the pathological affinity of –1.0 can only be reached when E_{fi} is zero for at least one topic; and this itself is a pathological input.
- Affinities are asymmetric. Although the agreement and disagreement metrics are symmetric, *f*'s affinities are driven by his own positions and emphases.
- The disagreement metric does not capture differences in sign; but sign still matters, because G_{fgi} is zero when the two groups' positions differ in sign.
- Strong disagreements do not yield indifference, that is, A_{fg} is not zero unless E_{fi} is 1.0 for the relevant topics.
- Zealots do distrust the lukewarm, provide that "zealot" is defined as a group that has a high Z_{fi} and a low E_{fi}.
- The lukewarm do admire the zealots. Affinity will be positive if signs are the same and E_{fi} is high on the topics for which group f is lukewarm and g is zealous.
- The ambivalent distrust the zealous, because D_{fgi} is included in the M_{fgi} term.
- A_{fg} is continuous for small changes in P_{fi} and E_{fi}, even (once the special cases are handled) as E_{fi} goes to 0.0.
- The value of A_{fg} is reasonable even when the number of topics is 1.

3.3 Computing Affinity

The formula for A_{fg} given in section 3.2.5.4 is substantially what we want, and behaves nicely for non-pathological inputs. Preventing division by zero and providing continuous outputs in the presence of pathological inputs requires special handling, as shown in this section.[5] All of what follows assumes the definitions made in Sections 3.1 and 3.2.2.

In general, special cases occur when E_{fi} is 0 for one or more topics, that is, when group *f*'s emphasis is wholly on disagreement. Following our original terminology, we refer to this as the "zero-tolerance" case: group *f* will not tolerate even the slightest disagreement on such a topic.

3.3.1 Definitions

We make the following additional definitions:

I	=	{all *i*}, the set of all topics shared by groups *f* and *g*.
J	=	$\{i \ni E_{fi} = 0\}$, the set of all topics for which group *f* has zero tolerance for disagreement.
K	=	$\{i \ni E_{fi} = 0$ and $P_{fi} \neq P_{gi}\}$, the subset of *J* for which entities *f* and *g* do not completely agree.
L	=	$\{ i \ni E_{fi} > 0\}$, the set of all topics for which group *f* does tolerate some disagreement. Note that $L = I - J$.

3.3.2 The Cases

Case A: $J = \emptyset$ and $\eta_{fg} + \sum_{i \in I} M_{fgi} = 0$

In this case,

$$A_{fg} = 0$$

Group *f* has no zero-tolerance topics, there is no commonality between *f* and *g*, and $Z_{fi} = Z_{gi} = 0$ for all *i*. In short, nobody cares about anything, so the affinity between *f* and *g* is zero.

Case B: $J \neq \emptyset, K = \emptyset, \eta_{fg} + \sum_{i \in J} Z_{fi} G_{fgi} + \sum_{i \in L} M_{fgi} = 0$

[5] See the memo "Mars Affinity Model", by William H. Duquette (whd12_002), 1 February 2012, for the derivation of these special cases.

In this case,

$$A_{fg} = 0$$

Group *f* has at least one zero-tolerance topic, but *f* and g agree completely on all such topics; there is no commonality between *f* and g; *f*'s zeal is zero on all topics; and because g agrees with *f* on all of the topics in *J*, g's zeal is also zero on them. In short, *f* has at least one "zero-tolerance" topic, and while it agrees completely with g on all such topics, it also doesn't care about any of them.

Case C: $J \neq \emptyset, K \neq \emptyset$

In this case,

$$A_{fg} = -1$$

This is the pathological "zero-tolerance" case. Group *f* tolerates no disagreement on at least one topic about which it and g disagree.

Case D: $J \neq \emptyset, K = \emptyset, \eta_{fg} + \sum_{i \in J} Z_{fi} \cdot G_{fgi} + \sum_{i \in L} M_{fgi} \neq 0$

In this case,

$$A_{fg} = \frac{\eta_{fg} + \sum_{i \in J} Z_{fi} \cdot G_{fgi} + \sum_{i \in L} M_{fgi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\eta_{fg} + \sum_{i \in J} Z_{fi} \cdot G_{fgi} + \sum_{i \in L} M_{fgi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

Group *f* has at least one zero-tolerance topic, but the two groups agree perfectly on all such; and the other terms are non-trivial.

Case E: $J = \emptyset$ and $\eta_{fg} + \sum_{i \in I} M_{fgi} \neq 0$

This is the nominal case; and in this case we can use the equation we derived in Section **Error! Reference source not found.**:

$$A_{fg} = \frac{\eta_{fg} + \sum_{i \in I} M_{fgi} \cdot \left(G_{fgi} - \beta_{fi} \cdot D_{fgi}\right)}{\eta_{fg} + \sum_{i \in I} M_{fgi} \cdot \left(1 + \beta_{fi} \cdot D_{fgi}\right)}$$

Note, however, that when $J = \emptyset$ the formula shown in Case D simplifies to this one. Consequently, the code can use the Case D formula for both cases D and E.

4. GENERALIZED REGIONAL ATTITUDE MODEL

The Generalized Regional Attitude Model (GRAM)[6] is a population dynamics model of the attitudes and behavior of groups within neighborhoods within the playbox. GRAM tracks changes in attitudes over time. Changes are driven by events and situations modeled within the client simulation (e.g., civilian casualties, presence of force units in a neighborhood, and so forth). The client simulation will usually use rule sets to analyze these *drivers* and provide attitude inputs to GRAM.

The effects of an attitude driver are not limited to the neighborhood and group that are directly affected by the driver. There are second order effects on other groups in the neighborhood, and on groups in other neighborhoods. These *indirect effects* generally weaken with distance; they can also be delayed in time.

As simulation time progresses, GRAM tracks the contribution of each driver to each attitude curve, thus enabling the significant drivers to be determined after the fact.

At present, GRAM supports two different kinds of attitude: the *satisfaction* of civilian groups with respect to various *concerns*, and the *cooperation* of civilian groups with force groups.

This section details the nature of attitude levels and curves, the kinds of inputs that can affect each, the spread of indirect effects across the playbox, and how the results are computed. The specifics of satisfaction and cooperation levels are described in detail.

GRAM is a generalization and extension of the JNEM Regional Analysis Model (JRAM), as developed for the Joint Non-kinetic Effects Model (JNEM), which was in turn based on an earlier model called the Regional Analysis Model, or RAM. RAM was developed for the National Simulation Center by the Texas A&M University's Department of Political Science, working with the George Bush School of Government and Public Service and the Texas Center for Applied Technology (both also at Texas A&M). RAM was part of the Spectrum Simulation to model biases, alliances, rivalries, and other aspects of inter-group relationships.

4.1 Attitude Curves

A group's satisfaction and cooperation levels are collectively referred to as the group's *attitudes*. Attitudes will change over time, depending on what happens to the group. Events and situations that affect a group's attitudes are called *attitude drivers*, or simply *drivers*. As an attitude changes over time, it traces out an *attitude curve,* $A(t)$; the value of the attitude at time 0 is denoted $A(0)$. The chosen unit of time is irrelevant to the model, but is usually decimal

[6] This discussion refers to GRAM V2, in which each civilian group resides in a single neighborhood. For a description of the earlier version, refer to this document in Mars V1.35 and prior.

days. The value of $A(t)$ may range from A_{min} to A_{max} (−100.0 to +100.0 for satisfaction curves, and 0.0 to 100.0 for cooperation curves).

More specifically, the satisfaction for group g with respect to concern c at time t is denoted $S_{gc}(t)$, and the cooperation for civilian group *f* with force group g at time t is denoted $\Omega_{fg}(t)$. The *initial satisfaction* is $S_{gc}(0)$ and the *initial cooperation* is $\Omega_{fg}(0)$.

Any attitude curve $A(t)$ is recomputed at a series of major time steps (tocks) t_1, t_2, t_3, ..., as follows:

$$\begin{aligned} A(t_1) &= A(0) + (\text{contributions from } 0 \text{ to } t_1) \\ A(t_2) &= A(t_0) + (\text{contributions from } t_1 \text{to } t_2) \\ &\vdots \end{aligned}$$

For simplicity in notation during the following discussion, we will use t_1 to denote the time of the current tock and t_0 to denote the time of the previous tock. The interval $(t_1 - t_0)$ is typically constant for all pair of tocks, but in principle the interval could vary from tock to tock. The "contributions" at each tock are due to attitude drivers and are of two kinds:

- The contribution of level effects
- The contribution of slope effects

The client simulation's rule sets analyze current drivers, producing level inputs and slope inputs on particular attitude curves. GRAM translates these inputs into level effects and slope effects as described in Section 4.5; these effects contribute to the attitude curves as described in Sections 4.1.1 and 4.1.2 respectively.

4.1.1 Level Effects

GRAM inputs can create *level effects* on attitude curves. A level effect is a nominal change of a specified magnitude which takes place over a specified period of time. The rate at which the effect's magnitude, or *limit*, is realized is defined by a *realization curve*. This section explains how to compute the nominal contribution of a particular level effect to an attitude curve during some time step.

4.1.1.1 Level Effects Defined

A level effect can be thought of as a tuple of the following elements:

The indices of the affected attitude curve (*gc* for satisfaction curves, *fg* for cooperation curves)

- *d*, the ID of the driver that resulted in this affect.
- *k*, an indicator of the effect's cause; see Section 4.1.5.
- *limit*, the nominal magnitude of the change
- *days*, the time interval in days over which the effect is realized

- ts, the effect's start time
- te, the effect's end time.
- τ, a parameter which controls the shape of the realization curve.

When working with a set of level effects, these parameters can be subscripted with the effect index *i*, e.g. $limit_i$ is the *limit* of the *i*th effect.

The realization curve for a level effect is defined by the following function $E(t)$:

$$E(t) = \begin{cases} 0 & t \le ts \\ limit \cdot \left(1.0 - e^{\frac{-(t-ts)}{\tau}}\right) & ts < t < te \\ limit & t \ge te \end{cases}$$

This exponential curve approaches but never actually reaches the *limit*. Consequently, τ is chosen that the curve is just ε from *limit* at time *te*:

$$\tau = \frac{days}{-\ln\left(\frac{\epsilon}{|limit|}\right)}$$

$E_i(t)$ denotes this function for a specific level effect *i*. The nominal contribution of level effect *i* at time step t_1 is therefore

$$\delta_i(t_{1,}t_0) = E_i(t_1) - E_i(t_0)$$

Note that those level effects with $te < t_0$ or $ts > t_1$ are guaranteed to contribute a value of 0 to the curve during the time step—they have either run their course before the interval starts, or have not begun to have an effect when the interval ends. For such effects,

$$\delta_i(t_{1,}t_0) = 0$$

This nominal contribution $\delta_i(t_{1,}t_0)$ is used to compute the actual contribution of effect *i* during the time step. In addition, as the simulation runs from time step to time step we accumulate the nominal contribution to date for each level effect *i*:

$$ncontrib_i(t_1) = ncontrib_i(t_0) + \delta_i\left(t_{1,}t_0\right)$$

Accumulating the nominal contribution to date for a level effect is useful because it allows us to watch the progress of the level effect in terms of the original inputs as the model runs forward in time.

4.1.1.2 Effect of Epsilon on Level Effects

GRAM uses an ϵ value[7], nominally equal to 0.01, that affects level effects in two ways:

- As described in Section 4.1.1.1, ϵ is used to calibrate the exponential curve so that it reaches ϵ of its *limit* at its end time, *te*.

- When scheduling level effects the requested realization time in *days* is ignored for effects whose $limit < \epsilon$. Instead, *te* is set to *ts* and *days* is set to 0, so that the effect makes its entire contribution at time step containing *ts*.

4.1.2 Slope Effects

GRAM inputs can create *slope effects*. A slope effect is an attitude change with a specified nominal slope (change/day). The effect will cause the attitude to change at that same nominal rate until the slope effect is terminated or reaches a threshold. This section explains how to compute the nominal contribution of a particular slope effect during some time step.

4.1.2.1 Slope Effects Defined

A slope effect can be thought of us a tuple of the following elements:

The indices of the affected attitude curve (*gc* for satisfaction curves, *fg* for cooperation curves).

- *d*, the ID of the driver that resulted in this effect
- *k*, an indicator of the effect's cause; see Section 4.1.5
- *slope*, the nominal change per day
- *ts*, the effect's start time in ticks
- *te*, the effect's end time in ticks

When working with a set of slope effects, these parameters can be subscripted with the effect index *i*, e.g., $slope_i$ is the *slope* of the *i*th effect.

It would seem that the nominal contribution of a slope effect *i* for time step t_1 is simply the *slope* times the duration of the time step:

$$\delta_i(t_{1,}t_0) = slope \cdot (t_1 - t_0)$$

However, the effect might not apply for the full time interval; indeed it might not apply for any of the time interval. There are the following cases:

[7] Defined in the model parameter database as `gram.epsilon`.

- $ts \geq t_1$, in which case the effect hasn't yet started to contribute to satisfaction.
- $te \leq t_0$, in which case the effect is no longer contributing to satisfaction.
- $ts < t_1$ and $te > t_0$, in which case the effect is contributing for some or all of the interval.

Consequently, we define

$$\delta_i(t_1, t_0) = \begin{cases} slope \cdot (\min(te, t_1) - \max(ts, t_0)) & \text{where } ts < t_1 \text{ and } te > t_0 \\ 0 & \text{otherwise} \end{cases}$$

As the simulation runs from time step to time step we accumulate the nominal contribution to date for each slope effect, just as we did for level effects:

$$ncontrib_i(t_1) = ncontrib_i(t_0) + \delta_i(t_{1,}t_0)$$

However, we must also consider *slope chains*.

4.1.2.2 Situations and Slope Chains

A slope chain is a sequence of slope effects related to a single driver, usually a *situation*[8], all of which target the same attitude curve. Chains are produced when a situation evolves over time, producing a sequence of slopes. The essential thing about effects in a slope chain is that they may not overlap in time, precisely because the chain really represents a single effect that happens to fluctuate over time.[9] (Section 4.5 describes the genesis of slope chains in detail). The following table represents a slope chain for some particular situation *d*. Note that all effects in the chain have the same *d*, curve indices, and *k*; these values are therefore omitted from the table:

TS	TE	SLOPE
7	24	5.2
24	39	0
39	108	2.5
108		7.1

The situation begins at tick 7, and the slope is 5.2. At tick 24 the situation becomes inactive, and the slope drops to 0. At tick 39 the situation becomes active again, with a slope of 2.5. At tick 108 things heat up and the slope rises to 7.1. Note that the final effect in the chain has no end time; slope chains are terminated by setting the slope to 0 in the final link in the chain.

---

[8]A *situation* is an on-going condition, known to the client simulation, that has satisfaction implications for as long as it lasts.

[9]And, in fact, it's implemented as a single effect, which contains a list of future start times and slopes.

The presence of slope chains affects the model in two ways. First, care must be taken when scheduling a new effect to update any exist chain. If the situation above were to change again at time 134, for example, the existing chain would need to be extended with the new slope.

Second, it is possible that several links in a chain will contribute to satisfaction during a single time step, e.g., if the slope changes several times during the time step. Since the links really represent a single fluctuating effect, the total nominal contribution of the links must be considered when computing the actual contribution of the chain.

Extending the equations described in the previous sections to account for slope chains would be a tedious exercise in notation. To summarize, then, the implementation must account for the following:

- A new link in a slope chain must terminate any previous link in that chain.
- The nominal contribution must be computed for the chain as a whole.
- If two or more links in a chain are active during a single time step, they must be treated as a single effect with respect to the computation of the actual contribution to the attitude curve.

4.1.2.3 Effects of Epsilon on Slope Effects

The same ϵ used in the scheduling and assessment of level effects is applied to slope effects in a different way. If, when a slope effect is being scheduled, its $slope < \epsilon$, it will treated as though its $slope = 0$. This prevents the computation of insignificant slope effects (and particularly of insignificant indirect effects) from affecting performance.

4.1.3 Ascending and Descending Thresholds

Every level and slope effect has, in addition to the elements shown above, an *ascending threshold* and a *descending threshold*, which are used to filter out effects when the nominal contribution is computed. These thresholds are denoted *athresh* and *dthresh*. The notion is that the given effect cannot increase the underlying curve A if $A \geq athresh$, and cannot decrease the underlying curve A if $A \leq dthresh$. Thus, a given satisfaction slope effect might have a slope of -5.0 and a *dthresh* of -20: the curve will lose 5 nominal points per day, but only up to a threshold of -20 satisfaction points. The effect is reasonably sharp, but cannot make the group more than mildly annoyed. On the other hand, so long as the effect is in place it may be difficult to improve the group's mood much above -20.

The handling of thresholds is straightforward: effects for which the underlying curve exceeds the relevant threshold are treated as though their nominal contribution were zero.
The thresholds are set when the effect is scheduled.

A judicious use of thresholds can greatly enrich the notion of the trend; see Section 4.3.4.

4.1.4 Scaling of Contributions

The actual contribution to any attitude curve $A(t)$should show the effects of diminishing returns (technically, *diminishing marginal utility*) as the extreme values are approached. Specifically:

- Positive contributions should have a stronger effect when $A(t)$ is near A_{min} and a weaker effect when $A(t)$ is near A_{max}.
- Negative contributions should have a stronger effect when $A(t)$ is near A_{max} and a weaker effect when $A(t)$ is near A_{min}.
- $A(t)$ should stay within the range A_{min} to A_{max} without being artificially clamped.

To achieve this, each non-zero *nominal contribution* to $A(t_1)$ is scaled given the value of $A(t_0)$, producing the *actual contribution*. The following scheme has the desired properties, provided that the total actual nominal contributions at each time step are less than $A_{max} - A_{min}$.
First, for each nominal contribution *ncontrib* to satisfaction curve $S(t_0)$, let

$$\text{scale}(ncontrib) = \begin{cases} 2 \cdot \dfrac{100 - S(t_0)}{200} \cdot ncontrib & \text{where } ncontrib \geq 0 \\[2ex] 2 \cdot \dfrac{100 + S(t_0)}{200} \cdot ncontrib & \text{where } ncontrib < 0 \end{cases}$$

With this formula, a nominal contribution of 10 points will cause the satisfaction level to move 10% of the distance from its current value toward the upper limit, +100, no matter what that current value actually is. Similarly, a nominal contribution of –10 points will cause the satisfaction level to move 10% of the distance from its current value toward the lower limit of –100. Consequently, nominal contributions can be thought of either as points *or* as percentage changes in the difference between the current value and the extreme.

Similarly, for each nominal contribution *ncontrib* to cooperation curve $\Omega(t_0)$, let

$$\text{scale}(ncontrib) = \begin{cases} \dfrac{100 - \Omega(t_0)}{100} \cdot ncontrib & \text{where } ncontrib \geq 0 \\[2ex] \dfrac{\Omega(t_0)}{100} \cdot ncontrib & \text{where } ncontrib < 0 \end{cases}$$

In the sections that follow, the notation "scale(x)" indicates that *x* has been scaled in this way.

4.1.5 Causes and Scaling

The discussions in Sections 4.1.1.1 and 4.1.2.1 show how to compute the nominal contribution of each level and slope effect during a given time step. Section 4.1.4 shows how the contributions can be scaled so that the attitude level does not get clamped at its maximum or minimum value. There is one more piece to the puzzle.

We could simply compute the actual contribution during the time step from t_0 to t_1 as follows:

$$acontrib(t_1) = \text{scale}\left(\sum_i \delta_i(t_{1,}t_0)\right)$$

This, however, presumes that the collection of level and slope effects that are active at a given time are all independent of one another, and that each should always contribute its full scaled magnitude. This is not necessarily the case. Level and slope effects are the result of independent drivers which affect the local civilian population and hence affect their attitudes. But even if the drivers are independent, the effects need not be.

People's capacity to respond to events and situations, their ability to feel horror and dismay on the one hand or joy and exultation on the other, can be saturated on a number of axes. Once their capacity is saturated due to drivers of a particular kind, further events of that kind occurring shortly thereafter are unlikely to have much additional effect. Consider, for example, a neighborhood which is experiencing a serious epidemic. It's unlikely that a second epidemic afflicting the neighborhood at the same time will change the results much.

GRAM handles this through the notion of *causes*. Each input to GRAM can be assigned a *cause*. Inputs due to similar drivers—e.g., bombings—will have identical causes. All effects stemming from a single input will share that input's cause.

When contributions to a single curve $A(t)$ share a single cause, their total contribution is the contribution of the largest effect. More precisely, for each cause *k* let I_k^+ be the set of effects *i* that have cause *k* and for which $\delta_i(t_{1,}t_0) > 0$; similarly, let I_k^- be the set of effects *i* that have cause *k* and for which $\delta_i(t_{1,}t_0) < 0$. The nominal contribution of the effects with cause *k* is then

$$\max_{i \in I_k^+} \delta_i\left(t_{1,}t_0\right) + \min_{i \in I_k^-} \delta_i\left(t_{1,}t_0\right)$$

If we treat GRAM inputs for which no cause is specified as having a unique cause *k*, then the attitude level at time step t_1 is

$$A(t_1) = A(t_0) + \text{scale}\left(\sum_k \max_{i \in I_k^+} \delta_i(t_{1,}t_0) + \min_{i \in I_k^-} \delta_i(t_{1,}t_0)\right)$$

We would then like to compute the actual contribution of each effect *i* to this final result. The scaled contribution of each level effect *i* with cause *k* is shared with the other effects with cause *k* that are active in that time step. The following equation allocates the total scaled contribution back to each of the constituent effects on a *pro rata* basis, resulting in the effect's *actual contribution to date*:

$$acontrib_i(t_1) = acontrib_i(t_0) + \begin{cases} scale\left(\dfrac{\delta_i(t_{1,}t_0)}{\sum_{j\in I_k^+}\delta_j(t_{1,}t_0)} \cdot \max_{j\in I_k^+}\delta_j(t_{1,}t_0)\right) & \text{if } i \in I_k^+ \\ scale\left(\dfrac{\delta_i(t_{1,}t_0)}{\sum_{j\in I_k^-}\delta_j(t_{1,}t_0)} \cdot \min_{j\in I_k^-}\delta_j(t_{1,}t_0)\right) & \text{if } i \in I_k^- \end{cases}$$

Accumulating the actual contributions to date is useful because it allows us to see precisely how the effect has changed $A(t)$—and ultimately, how a given driver has changed attitudes in general.

4.2 Neighborhoods and Groups

Having discussed attitude curves in general, it's now time to discuss satisfaction and cooperation curves in specific. But first, we must discuss neighborhoods and groups.

GRAM models attitudes in a geographical region called the *playbox*. The playbox is divided into N_n areas, which are called *neighborhoods*. Similarly, the civilian population of the playbox is divided into N_g groups. Each group g resides in a particular neighborhood *n.*

GRAM tracks satisfaction for each civilian group, and also tracks cooperation for each civilian group with each force group.

4.3 Satisfaction

GRAM tracks the satisfaction of civilian groups with respect to each of the groups' concerns. Summary statistics are computed for each group and each neighborhood.

4.3.1 Satisfaction Levels

The degree to which a group is satisfied with its condition with respect to some concern is described by a satisfaction value, sometimes called a *satisfaction level*. A satisfaction value is a decimal number S, where $-100.0 \leq S \leq +100.0$. A value of +100 denotes perfect satisfaction, and a value of -100.0 denotes utter dissatisfaction. The following rating scale is frequently used:[10]

SYMBOL	MEANING	MIDPOINT	RANGE
VS	Very Satisfied	80.0	$60.0 \le S \le 100.0$
S	Satisfied	40.0	$20.0 \le S < 60.0$
A	Ambivalent	0.0	$-20.0 \le S < 20.0$
D	Dissatisfied	-40.0	$-60.0 \le S < -20.0$
VD	Very Dissatisfied	-80.0	$-100.0 \le S < -60.0$

Note that this is a relative scale. Satisfaction is measured on several axes, and different groups place different weights on different axes. Thus, satisfaction values must generally be weighted in order to be comparable. See the discussion of saliency in Section 4.3.3.

4.3.2 Concerns

A group may be satisfied or dissatisfied along several different axes, e.g., personal safety, quality of life, and so forth. In GRAM these axes are called *concerns*. GRAM itself doesn't care what the concerns are; the client simulation is free to define any concerns it chooses. However, there is a standard set of civilian concerns that has generally been used over the last several years:

Autonomy (AUT): Does the group feel it can maintain order and govern itself with a stable government and a viable economy?

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

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

Quality of Life (QOL): QOL includes the physical plants that provide services, including water, power, public transportation, commercial markets, hospitals, etc., and those things associated with these services, such as sanitation, health, education, employment, food, clothing, and shelter.

[10] This scale is implemented by the simlib(n) type `qsat`.

4.3.3 Composite Satisfaction, Weights, and Saliencies

The satisfaction of group g with respect to concern c is denoted S_{gc}. It is frequently desirable to summarize the full S_{gc} matrix using a variety of weighted averages, collectively termed *composite satisfaction* values. For example, we are often interested in group g's composite satisfaction over all concerns; this is called the group's *mood*, and is denoted S_g.

We could compute such composites using a simple average, but that would neglect two important factors. First of all, a group usually places more importance on some concerns than others, and different groups often place importance on different concerns. The importance[11] a group places on a concern is called the group's *saliency* for the concern. Saliency is represented by a number L_{gc} where $0.0 \leq L_{gc} \leq 1.0$. The following rating scale is used:

SYMBOL	MEANING	VALUE
CR	Crucial	1.00
VI	Very Important	0.85
I	Important	0.70
LI	Less Important	0.55
UN	Unimportant	0.40
NG	Negligible	0.00

When averaging across concerns we must weight by the saliency of each. Note that this is an absolute scale.

Second, different groups are of different sizes, and some groups have more importance to the wider community than other groups. To a first approximation, the importance of a group will be proportional to its size; hence, we weight by the population of each group, $population_g$.[12] in that the size of a neighborhood group will affect its importance to the group as a whole; however, a small elite group can have a higher weight than a large underclass.

---

11 Gurr defines *salience* as "the strength of motivation to attain or maintain the desired value position" in *Why Men Rebel*, Ted Gurr, Princeton, NJ: Princeton University Press. 1970, p. 66. This book was a seminal source for the original RAM model.

12 Note that small elites can matter out of proportion to their size. Thus, we used to use a "rollup weight" instead of the actual population; the rollup weight was intended to be based on the population, but be modified for special cases. Experience reveals that the users usually left the rollup weight set at its default value of 1.0, so that not even population was being taken into account. Consequently, we are now using population explicitly, and will add an *elite factor* in as a separate term if and when it becomes necessary.

It has been shown[13] that the following equation properly "rolls up" satisfaction across any set A of groups and concerns:

$$S_A = \frac{\sum_A population_g \cdot L_{gc} \cdot S_{gc}}{\sum_A population_g \cdot L_{gc}}$$

Given this, we can define the following useful composite satisfactions.

The mood of each group g (note that the population term drops out):

$$S_g = \frac{\sum_c L_{gc} \cdot S_{gc}}{\sum_c L_{gc}}$$

The mood of each neighborhood *n:*

$$S_n = \begin{cases} \dfrac{\sum_{g \in n} population_g \cdot L_{gc} \cdot S_{gc}}{\sum_{g \in n} population_g \cdot L_{gc}} & \text{when } \sum_{g \in n} population_g \cdot L_{gc} > 0 \\ 0 & \text{otherwise} \end{cases}$$

4.3.4 Trends

A *trend* (also known as a *long-term trend*) is a systematic effect on an attitude curve. Earlier versions of GRAM explicitly defined the long-term trend as a slope effect producing a steady increase or decrease in a satisfaction curve over time. Current versions of GRAM simply allow the client simulation to define long-term trend effects on their own; the trend then becomes just another driver.

Thresholds (Section 4.1.3) make trends much more useful, because any slope effect, if applied over a long period of time, will drive an attitude curve to its minimum or maximum extreme. Thresholds limit trends to a more reasonable effect.

Trends can be even more useful when implemented in pairs. For example, regression to a mean (0.0, say) can be implemented by creating a positive slope effect with an *athresh* of 0.0 and a negative slope effect with a *dthresh* of 0.0. When the curve is negative, the first will drive it up to 0.0; when it is negative, the second will drive it down to 0.0.

See Section 4.1.2 for more information about slope effects, and Section 4.5 for more information about drivers.

[13]"Satisfaction Roll-Up", Robert G. Chamberlain, September 12, 2006

4.4 Cooperation

GRAM tracks the *cooperation* of civilian groups with force groups. Cooperation is a concept that derives from the TRADOC HUMINT methodology; it indicates the likelihood that one group *f* will provide intelligence to another group *g*. Note that cooperation is distinct from providing active aid to another group; this is sometimes called *collaboration*. GRAM does not model collaboration.[14]

4.4.1 Cooperation Levels

The probability that a member of civilian group *f* will provide intelligence to a member of force group *g* is expressed as a number between 0 and 100 called the *cooperation level* of *f* with *g* and is denoted Ω_{fg}. Note that the information flow is always from group *f* to group *g*, i.e., from the first group to the second.

Cooperation levels are often represented symbolically, as indicated in the following table:

SYMBOL	MEANING	VALUE	RANGE
AC	Always Cooperative	100.0	$99.9 \le \Omega \le 100.0$
VC	Very Cooperative	90.0	$80.0 \le \Omega < 99.9$
C	Cooperative	70.0	$60.0 \le \Omega < 80.0$
MC	Marginally Cooperative	50.0	$40.0 \le \Omega < 60.0$
U	Uncooperative	30.0	$20.0 \le \Omega < 40.0$
VU	Very Uncooperative	10.0	$1.0 \le \Omega < 20.0$
NC	Never Cooperative	0.0	$0.0 \le \Omega < 1.0$

4.4.2 Composite Cooperation

The average cooperation of all of the groups in neighborhood *n* with force group *g* is denoted Ω_{ng}, and is computed as follows:

$$\Omega_{ng} = \begin{cases} \dfrac{\sum_{f \in n} population_f \times \Omega_{fg}}{\sum_{f \in n} population_f} & \text{when } \sum_{f \in n} population_f > 0 \\ 0 & \text{otherwise} \end{cases}$$

[14] Ambassador Terry McNamara has pointed out that "collaboration" is a loaded word, and that a more neutral term is preferable.

We weight cooperation by the size of the population. The cooperation of group *f* with group g can be thought of as the probability that an arbitrary member of group *f* will give intelligence to group g. Thus, when averaging across groups it makes sense to weight by the number of members in each. If the population of the neighborhood is zero, then Ω_{ng} is 0.

4.5 Drivers, Inputs, and Effects

The client simulation analyzes attitude drivers (events and situations) and produces satisfaction and cooperation inputs for GRAM. These inputs create effects across the playbox; the distribution and relative magnitudes of these effects is called the *spread* of the input. This section explains how spread is computed and each input is turned into a collection of effects.

Each GRAM input specifies a direct effect on a particular attitude curve. This direct effect is a level or slope effect as described in Sections 4.1.1 and 4.1.2. The direct effect then engenders indirect effects on other attitude curves.

Each input optionally includes an ascending threshold (*athresh*) and descending threshold (*dthresh*) as described in Section 4.1.3. If not specified, these thresholds default to the minimum and maximum of the curve's range, i.e., the default *athresh* and *dthresh* for satisfaction inputs are +100.0 and -100.0. Each effect engendered by the input inherits the input's thresholds. This, incidentally, is why both thresholds are required. An *athresh* is only binding on a positive effect—but a positive input can produce both positive and negative effects.

Indirect effects differ from the direct effect that engenders them in two ways. First, their magnitude (*limit* for level effects, *slope* for slope effects) is affected by the relationship between the group directly affected and the group indirectly affected, and by the relationship between the neighborhoods in which they reside. Second, indirect effects in other neighborhoods can be delayed by an interval that depends on the two neighborhoods.

This section explains how to determine the magnitude and delay for each indirect effect. Before we can define them, however, we must first define the proximity and effects delay between two neighborhoods.

4.5.1 Neighborhood Proximities

Whether or not a GRAM input in neighborhood *n* has indirect effects in another neighborhood *m* depends in part of the proximity of neighborhood *m* to neighborhood *n*, and on the nature of the driver. For example, combat in a neighborhood will likely concern residents in nearby neighborhoods, and might even concern residents in far away neighborhoods. Other situations, such as a sewage spill, might only concern residents of the neighborhood in which the spill occurs.

Neighborhood proximity is defined by the $proximity_{mn}$ matrix, which defines the proximity of neighborhood n to neighborhood m from the point of view of residents of m. Each element of the matrix must have one of the following values:

VALUE	SYMBOL	NOTES
0	HERE	$m = n$; indirect effects always occur.
1	NEAR	n is near m; indirect effects are common.
2	FAR	n is far from m; indirect effects are rare.
3	REMOTE	n is remote from m; indirect effects never occur.

Note that proximity, in this sense, is not simply a measure of miles walked or driven, but rather a perception on the part of the residents of neighborhood m. For example, neighborhoods A and B might be adjacent on the map, but have a natural boundary (e.g., a river) between them, such that residents of A seldom travel to B; thus, residents of A might consider B to be FAR. Neighborhood C might be farther from A than B is, but C contains a popular destination, e.g., a shopping district, or an industrial district that employs many of the residents of A. The residents of A would then consider C to be NEAR, even though it's farther away than B, which is considered to be FAR.

Also, note that $proximity_{mn}$ need not be symmetric; it is okay if $proximity_{mn} \neq proximity_{nm}$. In our example, the residents of A frequently visit C and so regard C as NEAR; but if the residents of C seldom visit A, they might regard A as FAR.

For convenience, we also define the following symbols:

$$
\begin{aligned}
here_{mn} &= \begin{cases} 1 & if\ proximity_{mn} = 0 \\ 0 & if\ proximity_{mn} \neq 0 \end{cases} \\
near_{mn} &= \begin{cases} 1 & if\ proximity_{mn} = 1 \\ 0 & if\ proximity_{mn} \neq 1 \end{cases} \\
far_{mn} &= \begin{cases} 1 & if\ proximity_{mn} = 2 \\ 0 & if\ proximity_{mn} \neq 2 \end{cases} \\
remote_{mn} &= \begin{cases} 1 & if\ proximity_{mn} = 3 \\ 0 & if\ proximity_{mn} \neq 3 \end{cases}
\end{aligned}
$$

Because each civilian group resides in a single neighborhood, we can reasonably talk about the proximity between a pair of civilian groups f and g, denoted $proximity_{fg}$, and can then go on to define $here_{fg}$, $near_{fg}$, and so forth.

4.5.2 Proximity Limits

Because a single GRAM input can produce a vast quantity of indirect effects, performance may become an issue in practice. GRAM implements the notion of the *proximity limit*[15] as a means to cope with GRAM performance issues during training exercises. The proximity limit may be set as indicated in the following table:

LIMIT	CONSEQUENCES
far	An input in neighborhood n may have indirect effects in n, in neighborhoods near n, and in neighborhoods far from n.
near	An input in neighborhood n may have indirect effects in n and neighborhoods near n. If the proximity limit is changed to "near", existing indirect effects in far neighborhoods will be terminated.
here	An input in neighborhood n will have indirect effects only in n. If the proximity limit is changed to "here", existing indirect effects in near and far neighborhoods will be terminated.
none	No indirect effects will be scheduled. If the proximity limit is changed to "none", all existing indirect effects will be terminated.

4.5.3 Neighborhood Effects Delay

When an event occurs in neighborhood n, the response to that event in another neighborhood m may be delayed: news takes time to spread. This delay is probably correlated with the distance between the two neighborhoods, but as with proximity geographic distance doesn't tell the whole story. Given cell phones and the Internet, communications between any two points can be nearly instantaneous, making distance moot; and yet, just because news *can* travel instantly doesn't mean that it does. Thus we define $delay_{mn}$ as the time delay in decimal days for an input in neighborhood n to have an indirect effect in neighborhood m. Note that $delay_{nn} = 0$ for all n; there is no delay within neighborhood n itself.

As with proximity, we define $delay_{fg}$ as the delay between the neighborhoods in which civilian groups f and g reside.

4.5.4 Satisfaction Influence

A civilian group g is said to influence another civilian group f to the extent that direct effects on group g have indirect effects on group f. Influence depends on the proximity between the two groups, and on the relationship between them.

---

[15] The proximity limit is found in the model parameter database as "`gram.proxlimit`".

The relationship between the groups, R_{fg}, is a number from from -1.0 to +1.0, where +1.0 indicates that *f* regards g as a perfect friend and -1.0 indicates that *f* regards g as a perfect enemy. The relationship between a group and itself is usually +1.0; the relationships between other groups typically range from -0.6 to +0.6. Note that if the relationship is 0.0, then group g will have no indirect effects at all on group *f*. Also, note that relationships need not be symmetric; R_{fg} need not equal R_{gf}. It is up the client simulation to specify the relationships; either as input data or via a model like MAM (Section 3).

The magnitude of g's influence on *f* is a multiplicative factor denoted si_{fg} and defined as follows:

$$si_{fg} = \begin{cases} R_{fg} & proximity_{fg} < \text{remote} \\ 0 & \text{otherwise} \end{cases}$$

Given a direct effect on group g with magnitude M, then the maximum indirect effect (in absolute terms) on some other group *f* is $si_{fg} \cdot M$. This ensures that the indirect effects are no bigger in absolute terms than direct effects, and they will usually be smaller.

Indirect effects are further constrained by the near and far factors associated with the satisfaction input; see Section 4.5.6.

4.5.5 Cooperation Influence

Cooperation influence is more complicated than satisfaction influence because each cooperation effect involves two groups rather than one, a civilian group and a force group. In previous versions of GRAM, a direct effect on the cooperation of civilian group *f* with force group *g* in neighborhood *n* could affect group *f*'s cooperation with every force group in every non-remote neighborhood, according to the relationship between force group *g* and the other force groups.

In the previous version of GRAM, each civilian group could reside in multiple neighborhoods, creating what were called *neighborhood groups*; two neighborhood groups belonging to the same top-level group would have a relationship *R* of 1.0 with each other. In this version, where each civilian group resides in a single neighborhood, groups that would have been neighborhood groups are now simply modeled as distinct groups with a strong relationship. In order to get the same pattern of indirect cooperation effects, then, we must have indirect effects on other civilian groups with strong relationships.

This results in a two-step process for determining cooperation influence: first we determine the set of affected civilian groups, and then we determine the set of affected force groups. A civilian group *e* can be affected by a direct cooperation effect on civilian group *f* if the proximity between groups *e* and *f* is not remote and group *e* has a strong enough relationship with group *f*, or, in other words, when $si_{ef} \geq CRL$, where CRL is the cooperation relationship limit.[16] Thus, a force group's treatment of *f* affects the attitudes of *f*'s close friends toward that same force group.

Then, the cooperation influence of force group g on force group h is simply the relationship between the two groups, R_{hg}. The indirect effects apply to all civilian groups e that meet the above requirement and all force groups h.

Thus, if g does something to antagonize *f*, thus decreasing *f*'s cooperation with g, then *f*'s cooperation also decreases with friends of g, and increases with enemies of g.[17] The same applies for *f*'s friends e.

4.5.6 Here, Near, and Far Factors

The influence factors determine the maximum indirect effect a direct effect may have on each neighborhood and group. However, different drivers differ in scope. For example, accumulation of garbage or raw sewage in the streets of a neighborhood is likely of concern only to the residents of that neighborhood, whereas combat in a neighborhood is almost certainly of concern to residents of near neighborhoods, and possibly of concern to those in far neighborhoods as well. Typically we assume that indirect effects are strongest HERE, decrease in NEAR neighborhoods, and decrease even more in FAR neighborhoods. This is controlled by three parameters associated with each satisfaction and cooperation input: the *here* factor, denoted *s*, the *near factor*, denoted *p*, and the *far factor*, denoted q. We require that

$$0.0 \leq q \leq p \leq s \leq 1.0$$

The values of *s*, *p,* and q can vary from input to input, even for a single driver; however, we usually define *s*, *p* and q to be the same for all inputs for each kind of driver. Moreover, *s* is usually 1.0.

4.5.7 Computing Spread

Given the terms defined in the preceding sections, we can now define the direct and indirect effects resulting from an input to GRAM. First, the direct effect is simply a level or slope effect as defined in Section 4.1. It will have a magnitude, which is called the *limit* for level effects and the *slope* for slope effects; here, we'll call it simply *M*.

The indirect effects *j* on civilian group *f* of a direct satisfaction effect *i* to civilian group *g* will have magnitude

---

[16] Model parameter: `gram.coopRelationshipLimit`, nominally 1.0.

[17] The civilian group's perception of the relationship between *h* and *g* is likely to be inaccurate. For, now, though, the actual relationship is what we have and we are using.

$$M_j = M_i \cdot si_{gf} \cdot (s \cdot here_{mn} + p \cdot near_{mn} + q \cdot far_{mn})$$

In other words, the direct magnitude is adjusted by the influence factor, and by *s*, *p,* and *q* as appropriate.

All indirect effects are delayed by $effects\_delay_{fg}$; thus, if the direct effect has start time ts_i and end time te_i, the indirect effects will have start and end time

$$ts_j = ts_i + effects\_delay_{fg}$$
$$te_j = te_i + effects\_delay_{fg}$$

Similarly, the indirect effects *j* on civilian group *e* and force group *h* of a direct cooperation effect *i* on civilian group *f* with force group *g* will have magnitude

$$M_j = M_i \cdot R_{hg} \cdot (s \cdot here_{ef} + p \cdot near_{ef} + q \cdot far_{ef})$$

The effects will be delayed in the same way as satisfaction effects.

Note that when scheduling slope effects, the bookkeeping associated with slope chains must also be done; see Section 4.1.2.2.

4.6 Dynamic Civilian Groups

In versions of GRAM prior to Mars 1.39, civilian groups resided in neighborhoods and did not move. In the real world, and especially in war zones, mass migration is common. Due to war, catastrophes, or simply a nomadic lifestyle, populations move and shift, and GRAM needs to allow them to take their attitudes with them. This section describes a set of operations added to GRAM in support of scenarios involving population movement.

4.6.1 Use Cases

This section describes the kinds of population movement we envision, based on discussions with NSC, BCTP, and TRISA personnel over the last few years.

- Civilians flee a neighborhood in fear for their lives, and regroup in another neighborhood (or neighborhoods).
- Displaced persons in a neighborhood are made unwelcome, and move along to another neighborhood.
- People from a group in one neighborhood trickle into another neighborhood over a period of days or weeks.

- Nomadic peoples travel from one neighborhood to another and back again over the course of the year.

- People engage in mass pilgrimages, and then return home.

Note that these use cases are not implemented in GRAM as such; they are the province of the client model, e.g., Athena. But GRAM now provides the infrastructure to support them.
The following scenarios are not yet supported by GRAM:

- First-wave/second-wave friction, i.e., members of group A come to neighborhood N in two successive waves, A1 and A2. (Practically speaking, A1 is probably in N at time 0; A2 comes to N seeking refuge.) Being essentially the same group, A1 and A2 would have a high relationship, which rules out significant friction. This scenario will require a model of dynamic relationships.

4.6.2 Dynamic Operations

The application must be able to move a group from one neighborhood to another, split out a new group from an existing group, and transfer population from one group to another (provided that the two groups share a common ancestor). The following portions of GRAM must be considered when performing these operations:

Group Relationships. How does the change affect the group's relationships with other groups?

Attitude Levels. How does the change affect the group's satisfaction and cooperation levels.

Pending Attitude Effects. The group likely has pending level and slope effects on its satisfaction and cooperation curves. How does the change affect these?

History. GRAM keeps a record of the contributions to each attitude curve at each time advance, along with other data. How do the changes in the group affect the recording of and access to historical data?

Existing Drivers. If a group moves or a new group is formed, how is it affected by existing attitude drivers in its neighborhood?

The Change as a Driver. The reason for the change to the group might be an attitude driver in its own right, and will therefore affect the group's attitudes.

4.6.3 Moving a Group

The first operation is moving an entire group *g* from neighborhood *m* to neighborhood *n*.

Rationale: It is unlikely that a large, established group will move from its home neighborhood in its entirety; usually some remnant would be left behind. But a small group of displaced persons might be forced to move *en masse* from one neighborhood to another.

Group Relationships	No change
Attitude Levels	No change
Pending Attitude Effects	Discarded. See Section 4.6.6.
History	The group retains its history.
Existing Drivers	The application must apply existing drivers to group *g* in neighborhood *n*. GRAM does not have enough information to do it automatically.
The Change as a Driver	If the movement itself is due to a driver, the application must apply the driver to group *g* once it has been moved.

4.6.4 Splitting a Group

The second operation is to split out a new group *g* from group *f*, placing *g* in neighborhood *n*. Then, *g* is said to be the *child* of *f*, and *f* to be the *parent* of *g*.

A group's ultimate parent is called its *ancestor*; the model often cares whether two groups have a common ancestor. For convenience, the groups created at initialization have no parent, but are considered their own ancestors.

Rationale: It will be common for combat, joblessness, etc., to drive people from their homes. We will split out new child groups to represent the displaced subset, so that they can take their attitudes with them.

Group Relationships	Copy parent's relationships.
Attitude Levels	Copy parent's attitude levels.
Pending Attitude Effects	Do not copy the parent's pending attitude effects. See Section 4.6.6.
History	History begins for the new group.
Existing Drivers	The application must apply existing drivers in neighborhood *n* to the new group *g*. GRAM does not have enough information to do it automatically.
The Change as a Driver	If the split itself is due to a driver, the application must apply the driver to group *g* once it has been created.

4.6.5 Transfer Population between Groups

The third operation is to transfer population from group *e* to group *f*. Note that *e* and *f* must have a common ancestor, so that they have the same relationships.

Rationale: First, when population shifts due to combat or other bad circumstances, the shift will often take place a little at a time. We don't want to have to split out a new group each time an increment of population moves from neighborhood *m* to neighborhood *n*. Second, displaced persons can move back home, back to the group they came from.[18] Third, nomads move from neighborhood to neighborhood over the course of the year. We can model this as movement of the entire group, or by splitting two or more groups out of a parent and moving population around.

Group Relationships	Are unaffected.
Attitude Levels	Group *e*'s attitudes are unchanged. Group *f*'s attitudes are the average of *e*'s and *f*'s before the population shift, weighted by the proportions of *e*'s and *f*'s personnel in *f* after the shift.
Pending Attitude Effects	Do not copy the parent's pending attitude effects. See Section 4.6.6.
History	Each group retains its history.
Existing Drivers	If group *f* was alive before the shift, the application must apply existing drivers whose effect is affected by the change in population; for example, coverage fractions might change. This is standard behavior already, however, and so nothing special need be done. If group *f* was dead before the shift, then existing drivers need to be applied to *f* as though it were a new group.
The Change as a Driver	If the shift itself is due to a driver, the application must apply the driver to groups *e* and *f* after the shift.

4.6.6 Pending Attitude Effects

Pending attitude effects are level and slope effects resulting from satisfaction and cooperation inputs. These effects are applied to the relevant attitude curves over time, possibly after a delay. As described in the previous sections, groups that move lose their pending attitude effects, and new child groups begin with no pending attitude effects. This section explains why this is a reasonable solution.

Pending attitude effects are due to attitude drivers, of which there are two kinds: situations and events. Situations generally give rise to slope effects, and events to level effects. We will consider them in turn.

[18] Until we have dynamic relationships, there's no benefit to making the returnees a separate group; and even when we do, it will sometimes be appropriate for the returnees to join their original group.

Situations are on-going circumstances in neighborhoods. A group is affected by whatever situations are present in its vicinity. Thus, a moved or newly created group needs to be affected by the situations in its new locale. The situations affecting a moved group in its former location, or a parent group, are not relevant.[19] Thus, pending situation effects can be discarded.

Events are one-time happenings whose attitude effects play out over (a usually rather short) period of time. At first glance it seems appropriate to let a moved or newly created group retain its event effects or those of its parent. At second glance, though, it becomes clear that event effects are of two types: those that affect the group purely because of its location, and those that affect the group because of the group it is. Level effects resulting from power outages are in the former category; level effects resulting from civilian casualties are in the latter category. It makes sense to retain the latter but not the former; and GRAM at present has no way to distinguish between the two categories. Thus, for simplicity we will discard the latter with the former. We can revisit this decision at a later time, if it becomes an issue.

For now, consequently, moved groups and newly create child groups will have no pending attitude effects until existing drivers are applied to them.

4.6.7 Dead Groups and Neighborhoods

In the past, GRAM has forbidden groups with zero population. As groups are split, as population is transferred from one group to another, and as population is lost due to attrition in the client simulation, however, zero population is likely to occur. Rather than artificially maintaining at least one person in each group, as we've done in the past, we'll do the following:

- A group with 0 population is said to be *dead*; other groups are *alive*.
- GRAM will check for dead groups at each time advance.
- A group that is dead will lose all of its pending attitude effects: there's no one left to care.
- A group that is dead will receive no new attitude effects.
- No attitude history will be recorded for dead groups.

If groups are allowed to have zero population, then neighborhoods can have zero population.

[19] A child group can be split out in the same neighborhood as its parent; in this case, it might be reasonable to let it copy its parent's pending situation effects. But the application of existing drivers will have the same net effect, and there's no reason to right special purpose code for this rare case.

This primarily affects the computation of neighborhood mood and of neighborhood cooperation with force groups, thus:

- The mood of a zero-population neighborhood is defined to be 0.
- The cooperation of a zero-population neighborhood with any force group is also defined to be 0.

4.7 History

The ultimate effect of any attitude driver (i.e., an event or situation) is remarkably hard to predict. Because of the effects of scaling and causal analysis (Section 4.1.5) the actual contributions of the driver depend on the current attitude levels, along with everything else going on at the same time. In order to assess the actual contributions of any given driver after the fact, then, GRAM preserves a history of the actual contributions of each driver on each satisfaction and cooperation curve at each time tick. This history table can be used to produce a list of the most significant drivers over a certain period of time, with respect to any particular group or neighborhood.

GRAM also preserves the history of each civilian group's population and neighborhood of residence at each time advance, as they are required for the computation of neighborhood statistics over time.

5. RELATIONSHIP MULTIPLIER FUNCTIONS

A client simulation's attitude rule sets will usually model many kinds of effects that depend on the relationships between acting groups and affected groups. The strength of the effect depends on the relationship R between two groups, as mediated by one of a number of *Relationship Multiplier Functions* (RMFs). Given a relationship value R, where $-1 \leq R \leq 1$, each RMF returns a multiplier r. This r is typically multiplied by the nominal magnitude of an attitude input to produce the actual input.

5.1 Nominal Relationships

At one time, RMFs were computed such that -1 <= r <= 1; the **Linear** RMF, for example, returned 1 for R = 1 and -1 for R = -1. Thus, applying an RMF to an attitude input reduced the size of the input for anything but extreme relationships. Moreover, extreme relationships simply aren't used:[20] the practical range is more like $-0.6 \leq R < 0.6$, Thus, applying any RMF but **Constant** to an attitude input was tantamount to reducing the change by a factor of about 0.6. As a result, the change magnitudes shown in the client simulation's rules were misleading. One would expect a nominal change of 10.0, for example, but the actual change would always be smaller. This made it more difficult for casual users to analyze the probable effect of rule firings, and also complicated the process of getting rule inputs from our subject matter experts.

Consequently, each of the RMFs depends on a parameter[21] called $R_{NOMINAL}$, where $0 < R_{NOMINAL} \leq 1$. This is the *nominal relationship*, the relationship that the subject matter experts should keep in mind when writing attitude rules. When $R = R_{NOMINAL}$, we expect the RMF to return 1.0 (or -1.0) and thus have no effect on the outcome. As a side effect, RMFs can return values greater than 1.0 and less than –1.0. An RMF can therefore weaken an input, strengthen an input, change its sign, or leave it unchanged, based on the relationship value R.

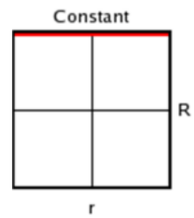
[20] Two groups with a relationship of 1.0 might as well be the same group; we usually only see values of 1.0 for the relationship of a group with itself. And a relationship of -1.0 is literally insane. If two groups A and B have a relationship of -1.0 then A is precisely as angered about something good that happens to B as B is pleased about it.

[21] Model parameter: `rmf.nominalRelationship`

5.2 Specific Relationship Multiplier Functions

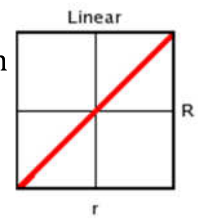
Mars defines the following RMFs:

Constant: The "Constant" function returns a constant 1.0, regardless of R. It is provided for use in generic code that takes the name of the RMF as an input, to remove the effect of group relationships.



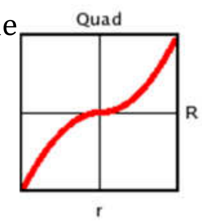
$$r = 1$$

Linear: The "Linear" function returns a value directly proportional to the relationship. Use this function when the sign and strength of an effect should match the sign and strength of the relationship. The resulting r will have a positive effect on friends and a negative effect on enemies, in proportion to the strength of the relationship.



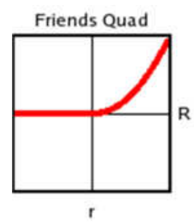
$$r = \frac{R}{R_{NOMINAL}}$$

Quad: The "Quad" function is similar in effect to the "Linear" function; however the resulting effect will be weaker than "Linear" for $|R| < R_{NOMINAL}$ and stronger than "Linear" for $|R| > R_{NOMINAL}$. It is computed as follows:



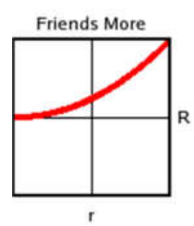
$$r = \left(\frac{R}{R_{NOMINAL}}\right)^2 \cdot \mathrm{sign}(R)$$

Friends Quad: The "Friends Quad" function has an effect that is strong for strong friendships, very weak for weak friendships, and zero for enemies of any degree. Its shape for friendly relationships is identical to the "Quad" function.



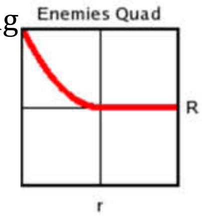
$$r = \begin{cases} \left(\dfrac{R}{R_{NOMINAL}}\right)^2 & R > 0 \\ 0 & R \leq 0 \end{cases}$$

Friends More: The "Friends More" function has a positive effect on both friends and enemies, but friends are affected much more strongly than enemies.



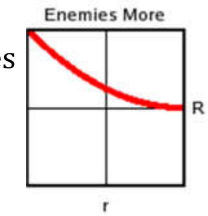
$$r = \left(\frac{1 + R}{1 + R_{NOMINAL}}\right)^2$$

Enemies Quad: The "Enemies Quad" function has an effect that is strong for strong enemy relationships, weak for weak enemy relationships, and zero for friends of any degree. Its shape for enemy relationships is similar to "Quad", but it does not reverse the sign.



$$r = \begin{cases} \left(\dfrac{R}{R_{NOMINAL}}\right)^2 & R < 0 \\ 0 & R \geq 0 \end{cases}$$

Enemies More: The "Enemies More" function affects both friends and enemies in the same direction, but friends are affected much less than enemies. Like "Enemies Quad", it does not reverse the sign.



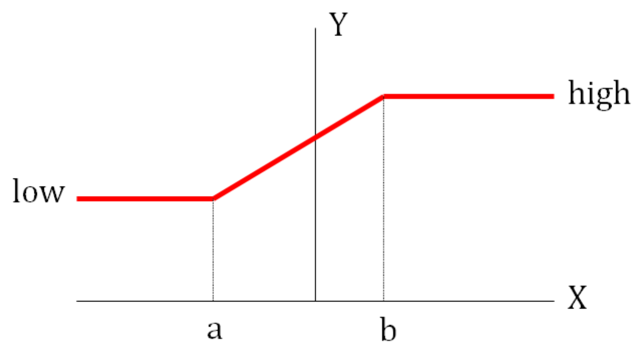
$$r = \left(\frac{1 - R}{1 + R_{NOMINAL}}\right)^2$$

6. MISCELLANEOUS MODELS AND ALGORITHMS

This section documents models and algorithms of general use.

6.1 Z-Curve Functions

A Z-curve is a stylized S-curve represented as a piece-wise linear curve of three segments. It is defined by the four parameters shown here:



In other words,

$$y = Z(x) = \begin{cases} low & \text{if } x \le a \\ low + \dfrac{x-a}{b-a}(high - low) & \text{if } a < x < b \\ high & \text{if } b \le x \end{cases}$$

6.2 Poisson Processes

In a Poisson process, the probability that a single event will occur during *any* very small interval of time is constant, whether other events have occurred recently or not. The average rate of occurrence, λ, determines that probability. The probability that n events will occur during an interval of length t is given by:

$$P_n(t) = \frac{e^{\lambda t}(\lambda t)^n}{n!} \text{ where } n = 0,1,2,3,\ldots$$

If the interval of time is always the same, the formula $\mu = \lambda t$ can be used to restate this formula for the probabilities recursively as follows:

$$\begin{aligned} P_0 &= \quad e^{-\mu} \\ P_n &= \quad P_{n-1} \cdot \tfrac{\mu}{n} \text{ where } n = 1,2,3,\ldots \end{aligned}$$

6.3 Selecting a Random Location in a Neighborhood

Use the following algorithm to select a random location in a neighborhood's polygon.

Let *tries* = 10.
While *tries* > 0,
    Select a point *p* randomly from the neighborhood's polygon's bounding box.
    If point *p* lies within the polygon, taking overlapping neighborhoods into account,
        Return point *p*.
        Done.
    Otherwise, decrement *tries*.
If no point has been found in 10 tries,
    Return the neighborhood's reference point.
    Done.

Appendices

7. ACRONYMS

AUT	Autonomy
BCTP	Battle Command Training Program
CTR	Contractor
CUL	Culture
GRAM	Generalized Regional Attitude Model
IGO	Inter-Governmental or International Organization
JNEM	Joint Non-kinetic Effects Model
JRAM	JNEM Regional Analysis Model
MAG	*Mars Analyst's Guide*
MAM	Mars Affinity Model
NGO	Non-Governmental Organization
NSC	National Simulation Center
QOL	Quality of Life
TRADOC	Training and Doctrine Command
TRISA	TRADOC Intelligence Support Activity
RAM	Regional Analysis Model
RMF	Relationship Multiplier Function
SFT	Safety