**Mars Analyst's Guide**

Mars Simulation Infrastructure Library, V2.10

October, 2012

William H. Duquette

[William.H.Duquette@jpl.nasa.gov](mailto:William.H.Duquette@jpl.nasa.gov)

Jet Propulsion Laboratory

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

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

Models 4

1. Introduction 5

1.1 Other Mars Documents 5

2. Mars Concepts 6

2.1 The Client Simulation 6

2.2 MAM: Modeling Belief Systems 6

2.3 GRAM: Modeling the Population 6

2.4 The Playbox 7

2.5 Groups **Error! Bookmark not defined.**

2.5.1 Civilian Groups 8

2.5.2 Force Groups 8

2.5.3 Organization Groups 9

2.6 Simulated Time 9

3. Mars Affinity Model 10

3.1 Belief Systems Defined 10

3.1.1 Topics 10

3.1.2 Positions 11

3.1.3 Emphasis 12

3.1.4 Belief Systems 12

3.2 Modeling Affinity 13

3.2.1 Desired Properties 13

3.2.2 Definitions 14

3.2.3 The Basic Model 15

3.2.4 Handling Ambivalence 16

3.2.5 Implicit Commonality 16

3.2.5.1 Adding Implicit Topics 17

3.2.5.2 Playbox Commonality 17

3.2.5.3 The Playbox Commonality Dial 18

3.2.5.4 Summary 19

3.2.6 Are the Properties Met? 20

3.3 Computing Affinity 21

3.3.1 Definitions 21

3.3.2 The Cases 22

4. Generalized Regional Attitude Model 24

4.1 Attitude Curves 36

4.1.1 Level Effects **Error! Bookmark not defined.**

4.1.1.1 Level Effects Defined **Error! Bookmark not defined.**

4.1.1.2 Effect of Epsilon on Level Effects **Error! Bookmark not defined.**

4.1.2 Slope Effects **Error! Bookmark not defined.**

4.1.2.1 Slope Effects Defined **Error! Bookmark not defined.**

4.1.2.2 Situations and Slope Chains **Error! Bookmark not defined.**

4.1.2.3 Effects of Epsilon on Slope Effects **Error! Bookmark not defined.**

4.1.3 Ascending and Descending Thresholds **Error! Bookmark not defined.**

4.1.4 Scaling of Contributions **Error! Bookmark not defined.**

4.1.5 Causes and Scaling **Error! Bookmark not defined.**

4.2 Neighborhoods and Groups **Error! Bookmark not defined.**

4.3 Satisfaction **Error! Bookmark not defined.**

4.3.1 Satisfaction Levels **Error! Bookmark not defined.**

4.3.2 Concerns **Error! Bookmark not defined.**

4.3.3 Composite Satisfaction, Weights, and Saliencies **Error! Bookmark not defined.**

4.3.4 Trends **Error! Bookmark not defined.**

4.4 Cooperation **Error! Bookmark not defined.**

4.4.1 Cooperation Levels **Error! Bookmark not defined.**

4.4.2 Composite Cooperation **Error! Bookmark not defined.**

4.5 Drivers, Inputs, and Effects **Error! Bookmark not defined.**

4.5.1 Neighborhood Proximities **Error! Bookmark not defined.**

4.5.2 Proximity Limits **Error! Bookmark not defined.**

4.5.3 Neighborhood Effects Delay **Error! Bookmark not defined.**

4.5.4 Satisfaction Influence **Error! Bookmark not defined.**

4.5.5 Cooperation Influence **Error! Bookmark not defined.**

4.5.6 Here, Near, and Far Factors **Error! Bookmark not defined.**

4.5.7 Computing Spread **Error! Bookmark not defined.**

4.6 Dynamic Civilian Groups **Error! Bookmark not defined.**

4.6.1 Use Cases **Error! Bookmark not defined.**

4.6.2 Dynamic Operations **Error! Bookmark not defined.**

4.6.3 Moving a Group **Error! Bookmark not defined.**

4.6.4 Splitting a Group **Error! Bookmark not defined.**

4.6.5 Transfer Population between Groups **Error! Bookmark not defined.**

4.6.6 Pending Attitude Effects **Error! Bookmark not defined.**

4.6.7 Dead Groups and Neighborhoods **Error! Bookmark not defined.**

4.7 History **Error! Bookmark not defined.**

5. Relationship Multiplier Functions 40

5.1 Nominal Relationships 40

5.2 Specific Relationship Multiplier Functions 41

6. Miscellaneous Models and Algorithms 43

6.1 Z-Curve Functions 43

6.2 Poisson Processes 43

6.3 Selecting a Random Location in a Neighborhood 44

Appendices 45

7. Acronyms 46

# Models

## Introduction

This document presents the models and related constructs implemented by version 2.10 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.

### 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](mailto:David.R.Hanks@jpl.nasa.gov) or [William.H.Duquette@jpl.nasa.gov](mailto:Willam.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 source tree in the form of software “man pages”.

## 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.

### 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*.

### 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.

### URAM: Modeling the Population

The Unified Regional Attitude Model (URAM) is a population dynamics model of the attitudes and behavior of groups within neighborhoods within the playbox. URAM 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 uses algorithms and rule sets to analyze these *drivers* and provide attitude inputs to URAM.

The effects of an attitude driver are not necessarily limited to the neighborhood and group that are directly affected by the driver—for certain kinds of attitude there may be second order effects on other groups in the neighborhood, and on groups in other neighborhoods. These *indirect effects* generally weaken with distance.

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

At present, URAM supports four different types of attitude curve: the *cooperation* of a civilian group with a force group, the *satisfaction* of a civilian group with respect to various *concerns*, the *horizontal relationship* of one group with another, and the *vertical* relationship of a group with an actor.

URAM has two major components: the URAM curve manager, which defines a general model and framework for attitude curves, and URAM proper, which defines the specific types of attitude curve listed above. The URAM curve manager is described in Section 4, and URAM proper is described in Section 0.

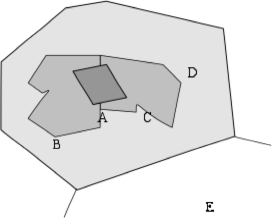
#### History

URAM is a generalization, extension and revision of the Generalized Regional Analysis Model (GRAM) developer for use by the Joint Non-kinetic Effects Model (JNEM) and by the Athena Stability and Recovery Operations simulation. GRAM was a generalization and extension of JRAM, also developed for 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.

URAM was designed specifically for use in version of the Athena S&RO simulation. It supports longer time horizons explicitly, and models variation in horizontal and vertical relationships as well as satisfaction and cooperation.

### The Playbox

URAM 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 URAM; 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]](#footnote-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.

### Groups

The people in the playbox are divided into *groups*, of which there are three kinds: civilian groups, force groups, and organization groups. URAM models the horizontal relationships (positive or negative) between pairs of groups as these relationships change over time in response to events and situations.

#### 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.

URAM models the satisfaction of civilian groups with respect to various concerns, and the cooperation level of civilian groups with force groups, tracking these attitudes as the group members are affected by events and situations taking place in the client simulation. These attitudes then will typically affect the groups' reactions and responses in the client simulation.

#### Force Groups

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

#### 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.

### Actors

In addition to groups, there are also *actors*: significant decision makers within the playbox. URAM models the vertical relationship (positive or negative) of groups with actors as these relationships change over time in response to events and situations.

### 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.

The URAM model, however, is designed explicitly for a time step of a week.

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.

## Mars Affinity Model

Athena and JNEM rely heavily on the concept of the horizontal relationship between two groups *f* and *g*, where . Athena also defines the vertical relationship between a group *g* and an actor *a*, where . 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 of the groups in a region, and then rate each group with respect to each of them, that we could use that as a basis to compute some notion of relationship between the two groups.

Given two groups *f* and *g*, then, we define the *affinity* of group *f* for group *g* to be the natural degree of relationship between the two groups given their particular *belief systems*. Affinity is denoted , where . 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.

### Belief Systems Defined

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

#### 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 is something about which *f* and *g* can have opinions that can be meaningfully compared.

#### Positions

Then, let be the *position* of group *f* with respect to topic *i*, where 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,

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 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]](#footnote-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 |

#### Emphasis

The positions 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 to be group *f’*s *emphasis* on agreement or disagreement with respect to topic *i*, where .

If 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 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 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]](#footnote-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 |

#### Belief Systems

The values of and for all *i* are said to constitute group *f’*s belief system.

### Modeling Affinity

#### Desired Properties

Our formula for computing should have the following properties:

* Relationships of –1.0 are pathological; 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, need not equal , 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, 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]](#footnote-4)
* should be generally continuous for small changes in any input.
* The value of should be reasonable even when the number of topics is 1.

#### Definitions

Given *N* topics numbered , and two groups *f* and *g*, let be group *f’*s position on topic *i*, and let 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:

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, , is defined as follows:

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:

* has the same “units” as .
* If or is 0.0, then there is no agreement; .
* If , then .
* 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 is 1.0, but 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, , is defined as the average difference in their positions:

Again, this has several useful properties:

* has the same “units” as ; thus, agreements and disagreements are comparable.
* Disagreement is symmetrical. (Affinity will not be.)

Next, we define the emphasis ratio:

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

#### 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:

The denominator normalizes the result so that .

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

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

* As mentioned above, is undefined when is 0.0; there are several distinct cases.
* The denominator will be 0.0 if is 0.0 for all *i*.

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

#### Handling Ambivalence

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

We used the ’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 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 to be the importance of topic *i* to group *f* in the context of group *g*’s belief system, where

In short, strong disagreements trump weak zeal. The equation for then becomes

or, equivalently,

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

#### 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.

##### 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 :

where is the number of implicit topics of agreement between *f* and *g*. If is 0, we have the same model we had before. As increases, so does the implicit commonality of the two groups. When , implicit commonality balances the explicit topics; when , the explicit topics dominate, and when , the implicit commonality dominates.

##### Playbox Commonality

It is clear that 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 as the cultural commonality within the playbox, and as group *f*’s share in that common culture, where . If group *f* belongs to the dominant culture in the playbox, then . If group *f* is a complete outsider, then . Then we can define as follows:

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, doesn’t make a good input, either; ideally it ought to depend on . So let

, for

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

##### The Playbox Commonality Dial

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

for

The input 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 will tend to increase or decrease affinities across the board, to the extent that the groups involved participate in the playbox commonality.

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

#### Relevance of Topics

In the real world, some topics have more relevance than others. In country X, for example, suppose that the groups in the region have strong beliefs about the desirability of U.S. intervention in that region. Some are strongly for it; others are strongly opposed. This is a significant fault line. Suppose, however, that such intervention is entirely hypothetical. The U.S. has not intervened in living memory, and no one in the country has any reason to expect that it might in the foreseeable future. In this case, it's unlikely that the topic "U.S. Intervention" is going to have a significant effect on affinities. The topic is not relevant.

If the U.S. announced that it was likely to bring troops to the country for some reason, then this topic would suddenly become relevant, and affinities would change as a result. We model this as follows.

First, let be the relevance of topic *i*, where . When topic *i* has a relevance of 0.0, it should have no effect on affinities; when it is 1.0, it is fully relevant, and has the effects described in the previous sections.

Next, let

This is group *f'*s position on topic *i* given the relevance of topic *i*. Then, let

When is 1.0, the model will return the same results as before; and as decreases, every entity’s position on topic *i* will decrease with it, reducing both agreements and disagreements. When is 0.0, both agreements and disagreements on topic *i* will be 0.0, and the topic will have no effect on the results.

Or rather, it *would* have no effect on the results if it weren’t for playbox commonality, which is defined as

for

where is the playbox commonality dial and *N* is the number of topics. The intent of the playbox commonality value is to balance the explicit topics, which tend to disagreement, with an equal number of implicit topics on which there is general agreement. Decreasing topic relevance effectively decreases the number of explicit topics. For example, a belief system with ten completely irrelevant explicit topics would result in highly positive affinities based on the playbox commonality’s implicit topics. Hence, playbox commonality must be redefined as follows:

This will yield the same number as before when is 1.0 for all *i*, and will decrease proportionately as relevance decreases, thus balancing the relevant topics and not the irrelevant topics.

#### Summary

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

Let

= Playbox commonality dial, where ; the value is nominally 1.0.

= Entity commonality dial, where ; nominally 1.0 for local entities and 0.0 for non-local entities.

= The relevance of topic *i*

Then

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

#### Are the Properties Met?

This model meets our desired properties.

* .
* 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 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 is zero when the two groups’ positions differ in sign.
* Strong disagreements do not yield indifference, that is, is not zero unless is 1.0 for the relevant topics.
* Zealots do distrust the lukewarm, provide that “zealot” is defined as a group that has a high and a low .
* The lukewarm do admire the zealots. Affinity will be positive if signs are the same and is high on the topics for which group *f* is lukewarm and *g* is zealous.
* The ambivalent distrust the zealous, because is included in the term.
* is continuous for small changes in and , even (once the special cases are handled) as goes to 0.0.
* The value of is reasonable even when the number of topics is 1.

### Computing Affinity

The formula for given in section 0 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]](#footnote-5) All of what follows assumes the definitions made in Sections 3.1 and 3.2.2.

In general, special cases occur when 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.

#### Definitions

We make the following additional definitions:

*I* = {all *i*}, the set of all topics shared by groups *f* and *g*.

*J* = {}, the set of all topics for which group *f* has zero tolerance for disagreement.

*K* = {}, the subset of *J* for which entities *f* and *g* do not completely agree.

*L* = {}, the set of all topics for which group *f* does tolerate some disagreement. Note that .

#### The Cases

**Case A:**

In this case,

Group *f* has no zero-tolerance topics, there is no commonality between *f* and *g*, and for all *i*. In short, nobody cares about anything, so the affinity between *f* and *g* is zero.

**Case B**:

In this case,

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:**

In this case,

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:**

In this case,

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:**

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

Note, however, that when 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.

### Congruence

Information operations campaigns (and advertising in general) appeal to the civilian population by appealing to their values, that is, to their beliefs. We say that messages of this kind rely on a *semantic hook* to catch their listeners. Such a semantic hook will consist of positions on a small number of topics, much smaller than the full set used to compute affinity. (Indeed, semantic hooks can include positions on topics that are not important enough to be used when computing affinity.)

By limiting our affinity equation to only those topics contained in the semantic hook, we can compute the *congruence* of the semantic hook with a group's belief system. Denoted , where , congruence is the affinity of the group for the positions expressed by the hook. Note that unlike standard affinity, *congruence* is not bidirectional:

* The congruence of the hook with the group is the group's affinity for the hook's positions.
* The hook not a group or an actor; it has no affinities of its own.

When semantic hooks are used to compute congruence, it is common that the belief system will include a number of topics that are only used for this purpose, independent of the topic relevance. Thus, every topic has an *affinity flag*, that indicates whether or not it should be used when computing group-to-group affinities.

#### Computing Congruence

Congruence is computed in the same way as affinity, with these differences:

* The group is group *f*.
* The hook stands in for group *g*.
* The algorithm doesn't depend on group *g*'s emphasis on agreement/disagreement, so the hook doesn't contain one.
* The algorithm looks only at the topics contained in the hook.
* The hook's entity commonality, , can be set according to the perceived sender of the message containing the hook, or 1.0 by default.

## The URAM Curve Manager

The section describes the URAM curve manager, and its notion of attitude curves. The URAM curve manager underlies URAM proper.

Version 3 of Athena contained two distinct kinds of attitude curves; and these kinds differed in their dynamic characteristics: satisfaction and cooperation curves on the one hand, and vertical relationship curves on the other. GRAM handled the one, and the Athena handled the other *ad hoc*. The URAM curve manager provides a framework that supports both kinds of curve, and offers many additional possibilities.

### The Attitude Equations

This section describes the mathematics behind URAM's notion of attitude curves, as implemented by ucurve(n), the URAM curve manager. It also explains how this notion grew beyond that of GRAM.

#### Classic Satisfaction and Cooperation Curves

The basic equation in classic GRAM is the following:

where

= The level of the attitude at time *t*.

= The current change in the attitude due to level and slope effects (i.e., rule set inputs)

That is, at each time step we compute the change in attitude due to level and slope effects, , and add it to the previous time step’s attitude to get the current attitude.

Note that in this model all attitude changes at this time step persist indefinitely: the value at the next time step is the value at this time step plus any deltas. This persistent character is a big problem for longer time scales; even very small slope effects can drive an attitude curve to its minimum or maximum value over the course of a few months.

Classic GRAM also supports magic attitude adjustments, which are essentially changes made directly to between time advances. Because they occur between time advances, they are not included in the equation shown above.

#### Vertical Relationship Curves

In Athena v3, vertical relationships are handled by the application rather than by GRAM, and the fundamental equations are different:

where

= The vertical relationship at time *t*

= The baseline vertical relationship at time *t*

= Transient deltas to the vertical relationship at time *t*

= The time at which control of the relevant neighborhood last shifted.

That is to say, at each time step the vertical relationship is the baseline value plus the current deltas, which are completely transient. The baseline only changes in response to particular events in the simulation, e.g., when a different actor gains control of a neighborhood.

Thus, we can remove the reference to by rewriting these equations as follows:

where is the sum of any adjustments made to the baseline at time *t*—and the baseline is only adjusted when control of the neighborhood shifts.

#### The Unified Equations

We can represent both of the classic patterns with a single set of equations. This section and those following build up the equations in several steps. First,

where

= The current value of the attitude at time *t*

= The baseline value of the attitude at time *t*

= The current deltas, due to input effects, at time *t* (see Section 4.4)

= The current level multiplier

= The baseline level multiplier

If we set to 1.0 and to 0.0, we get the GRAM equations; if we set to 0.0 and to 1.0 we get the Athena vertical relationship equations. Now, let us add some constraints on and :

.0

.0

The equation can now be interpreted as an exponential smoothing equation, with as the smoothing coefficient: the new baseline is a moving average of the current value with past history, which is represented by the old baseline. Thus, we can see that in GRAM we throw away past history when computing the baseline; only the current level matters. With vertical relationships we only look at past history when computing the baseline. But we can easily pick values of in between 0.0 and 1.0 and thus keep varying amounts of the history. And this is exactly what we want to do.

#### Regression to a Natural Level

In GRAM we can use trends and thresholds to make a curve regress to some desired natural level. The mechanism is clumsy and difficult to use. We can add a similar though much simpler capability to our unified equations, as follows:

where

= The natural level of the attitude at time *t*

= The natural level multiplier

and

Just as pulls the baseline toward the current level, so pulls the baseline toward the natural level. If both and are non-zero, with larger than , then the baseline will move towards when is large, but towards when is small.

The value of can be set at time 0 and left alone; or it can be recomputed at each time step according to some external model. For example, the natural level of a group’s SFT satisfaction is naturally some function of the group’s security. As the group’s security increases and decreases, so should increase and decrease.

If an attitude type has no natural level, then can be set to 0.0 and the value of is irrelevant. We suggest that be set to 0.0 in such cases.

#### Persistent and Transient Deltas

Because the is added to the current level,, rather than to the baseline, it is said to be transient—it only persists to the extent that the factor rolls into . We can also support persistent changes, i.e., changes made directly to the baseline. The new equations look like this:

where

= Persistent deltas due to input effects

= Nominal baseline at time *t*, before persistent deltas are applied.

We introduce the term because of scaling; persistent effects will be scaled with respect to , while transient effects will be scaled with respect to . See Section 4.4 for more information.

#### Clamping

The and terms can be arbitrarily large, even with scaling, leading to results that are out-of-bounds for the attitude. Thus, we need to clamp the results within the [*min*, *max*] range for the attitude typewhen computing and . Thus, the final statement of the equations is the following:

where the clamp(*x*) function returns:

* *min* if *x* is less than *min*,
* *max* if *x* is greater than *max*,
* and *x* otherwise.

#### The Complete Equations

Putting it all together, we have the following:

where

= The current value of the attitude at time *t*

= The baseline value of the attitude at time *t*

= The nominal baseline at time *t*, before persistent deltas are applied.

= The natural level of the attitude at time *t*

= Transient deltas due to input effects at time *t*

= Persistent deltas due to input effects at time *t*

= The current level multiplier

= The baseline level multiplier

= The natural level multiplier

and

### Attitude Types

In principle, an attitude curve is defined by eight parameters:

*min* = The minimum curve value

*max* = The maximum curve value

*α* = The current level multiplier

*β* = The baseline multiplier

*γ* = The natural level multiplier

= The current level of the attitude at time 0

= The baseline level of the attitude at time 0

= The natural level of the attitude at time 0

which are subject to the following constraints:

Rather than defining all eight of these parameters for each attitude curve, we find it convenient to group the existing attitude curves into types (e.g., satisfaction with the four concerns, horizontal relationships, vertical relationships, and cooperation levels). In this scheme, the type defines the *min*, *max*, *α*, *β*, and *γ* parameters, and the individual curve has its own , , and parameters.

The URAM Curve Manager does not itself define any specific curve types; that is left to the client (i.e., URAM proper).

### Baseline Adjustments

A baseline adjustment[[6]](#footnote-6) is an absolute, unscaled step change to the baseline of the curve caused by some attitude driver. Adjustments are applied between time advances, and affect the value of directly. The adjustment delta is ascribed to the driver at the next time advance (See Section 4.6).

It is unusual to have more than one adjustment on a curve in a single time step.

### Transient Effects

A transient effect represents how a particular attitude driver affects an attitude during this time step, i.e., it is a contribution to . For example, my quality-of-life might be lower than usual right now because the power is out. My QOL satisfaction might change by -5.0 points, and this effect will remain so long as the power remains off. When the power comes back on the effect will disappear. (Whether or not the baseline QOL value will change as a result of the power outage depends on the value of used with QOL curves.) This is a transient effect.

Unlike GRAM slope effects, which remained in GRAM and continued to have an effect time-step-by-time-step until they were explicitly terminated, URAM’s effects apply to a single time step and then disappear. If the attitude driver (i.e., a situation) persists across time steps and should have a transient effect in each time step, it must create a new transient effect in URAM at each time step.

Transient effects are much more complicated than adjustments, as they inherit the cause-and-scaling model from GRAM. On the other hand, they are simpler than GRAM’s level and slope effects. Each effect is defined by the following attributes:

* The magnitude, expressed as some number of nominal percentage points of change.
* The cause, expressed as an integer ID.[[7]](#footnote-7)

We will discuss each of these in turn.

#### Scaling of Contributions

The actual contribution of a transient effect to an attitude curve 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 is near *min* and a weaker effect when is near *max*.
* Negative contributions should have a stronger effect when is near *max* and a weaker effect when is near *min*.
* Ideally, should stay within the range without being artificially clamped.

To achieve this, we scale each nominal contribution to given the current value of , producing the actual contribution. The following function has the desired properties provided that the total unscaled contributions at each time step are less than 100.0 in absolute terms. (If they are greater, the value of will be clamped.)

First, the nominal magnitude *M* of a transient effect is a number representing a percentage change in the attitude relative to baseline *B*. The function computes the actual magnitude , which is the delta that produces the percentage change in *B*. The *B*, *min*, and *max* parameters depend on the attitude curve for which scaling is being computed and can be known from context; hence, although the function would more properly be written “scale(*M, B, min, max*)” we will usually write it “scale(*M*)”.

For example, suppose that *M* is 10.0. This represents a 10% change from the baseline to *max*. The following table shows the actual contribution of a nominal 10.0 point change for a curve with bounds (-100.0, +100.0) and several different baselines.

|  |  |  |
| --- | --- | --- |
|  | *M* |  |
| -100.0 | 10.0 | 20.0 |
| -50.0 | 10.0 | 15.0 |
| 0.0 | 10.0 | 10.0 |
| +50.0 | 10.0 | 5.0 |
| +100.0 | 10.0 | 0.0 |

#### Causes and Scaling

Section 4.4.1 shows how to compute the scaled contribution to of each transient effect during a given time step. There is one more piece to the puzzle.

We could simply compute the contribution of the transient effects during time step *t* as follows:

where

= The nominal magnitude of the *i*th transient effect

This, however, presumes that the transient effects at time *t* are all independent of one another, and that each should always contribute its full scaled magnitude. This is not necessarily the case. Transient effects are the result of independent drivers that 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 that is experiencing a serious epidemic. It's unlikely that a second epidemic afflicting the neighborhood—or the indirect effects of an epidemic in the neighborhood next door—will change the results much. The first epidemic does whatever damage will be done.

The URAM Curve Manager handles this through the notion of *causes*. Each transient effect is assigned a *cause*. Effects due to similar drivers—e.g., epidemics—will have identical causes.

When effects on curve share a single cause, their total contribution is the contribution of the largest effect. More precisely, for each cause *k* let be the set of effects *i* that have cause *k* and for which ; similarly, let be the set of effects *i* that have cause *k* and for which *.* The nominal contribution of the effects with cause *k* is then

We can then say that

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 during time step *t*:

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

### Persistent Effects

Some attitude drivers, e.g., power outages are essentially transient. While the power is out, it is a problem; when the power comes back on, the problem goes away. Other attitude drivers, however, cause a one-time, persistent change in an attitude. When control of a neighborhood shifts from one actor to another, for example, there is a persistent one-time change in satisfaction.

Persistent effects are similar to transient effects, and their contributions are computed in essentially the way. The distinctions are as follows:

* Persistent effects contribute to rather than .
* Persistent effects are scaled with respect to rather than with respect to .

### Historical Data

Every adjustment and effect is generated by some attitude driver known to the client. In order to understand causality, we need to relate drivers to their contributions. Consequently, at each time *t* we will save the total contribution of driver *d* to each curve for later retrieval. For baseline adjustments, the contribution is simply the value of the adjustment; for persistent and transient effects, it is as computed in Section 4.4.2.

### Applying Adjustments and Effects

When time is advanced, all pending adjustments and effects must be applied and the curve’s and values computed. The algorithm is as follows:

Update based on external models, if need be.

Save the contributions of baseline adjustments by driver.[[8]](#footnote-8)

Compute , along with the resulting positive and negative scaling factors.

Compute given any pending persistent effects.

Compute , along with the resulting positive and negative scaling factors.

Compute given any pending transient effects.

Compute .

Save the contributions of all persistent and transient effects by driver.

#### Applying Transient Effects Only

As a practical matter the initial input to URAM will include , the initial baseline level, but not , the initial current level, because depends on transient effects, which in turn depend on simulated attitude drivers. At *t*=0, then, we will need to apply transient effects based on the initial state of the simulation. At the same time, we are given ; if we apply the algorithm shown just above, we will recompute , which we do not want to do. Thus, at *t*=0 we will compute by applying transient effects only, using the following algorithm:

Save the contributions of baseline adjustments by driver.[[9]](#footnote-9)

Compute the positive and negative scaling factors for .

Compute given any pending transient effects.

Compute .

Save the contributions of all transient effects by driver.

### Examples

This section shows graphically the effect of different settings of α, β, and γ on transient effects. Consider a satisfaction Quality-of-Life (QOL) curve, and a driver that causes a −10.0 nominal change each week for four weeks. With our classic model (, this results in the following graph (assuming that QOL began at 0.0):

The actual and baseline levels drop by 10.0 points each week, leaving satisfaction at −40.0 after four days. The driver ends, and in the absence of other drivers the curve simply sits there at −40.0. If we change to 0.9, (that is, ) we get this instead:

Satisfaction drops immediately, and stays approximately 10.0 points worse than before for all four weeks. After that, the effect ends, and things are immediately about 10.0 points better—but not completely. The negative transient effect has dragged the baseline down by about 4.0 points over the four weeks, and and so the group is less satisfied with QOL than before the driver occurred. Note especially that while the classic model ran away very quickly, the new model does not.

The above graphs presume that the curve has no natural level, i.e., γ is 0.0. Assume that the curve has a natural level of , and set to 0.8 and to 0.1, leaving at 0.1. The resulting curve looks almost identical to the previous…but the curve slowly trends back toward 0.0:

The speed of the regression depends on the distance between and *C*.

## The URAM Curve Types

The Unified Regional Attitude Model (URAM) is introduced in Section 2.3; its notion of an attitude curve is described in Section 4. This section describes URAM proper, the four curve types (cooperation, horizontal relationship, satisfaction, and vertical relationship), and all of their particular details.

### Neighborhoods, Groups, and Actors

URAM models attitudes in a geographic region called the *playbox*. The playbox is divided into areas which are called *neighborhoods*. The civilian population of the playbox is divided into *civilian groups*; each group *g* resides in a particular neighborhood *n*. Any number of groups may reside in a single neighborhood, but a group only resides in one neighborhood.

The significant decision makers in the playbox are called *actors*. An actor may be an individual, a cabal, a political party, the leadership of a particular group or collection of groups, or an entire country.

An actor can own *force groups* and *organization groups*, organized collections of personnel that the actor can use to carry out his tactics.

These entities are only fully realized in the client simulation; in URAM we are concerned only with their attitudes, and especially with their attitudes toward each other.

### Drivers, Inputs, and Effects

The client simulation analyzes attitude drivers (events and situations) and produces attitude inputs for URAM. Satisfaction and cooperation 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 URAM input specifies a direct effect on a particular attitude curve. This direct effect is a persistent or transient effect as described in Sections 4.4 and 4.5. For satisfaction and cooperation curves, the direct effect then engenders indirect effects on other attitude curves. [[10]](#footnote-10)

Indirect effects differ in magnitude from the direct effect that engenders because of the relationship between the group directly affected and the group indirectly affected and proximity between the neighborhoods in which they reside.

This section explains how to determine the magnitude for each indirect effect. Before we can define them, however, we must first define some terms.

#### Influence

An attitude curve is said to influence another attitude curve of the same type to the extent that direct effects on the first curve will have indirect effects on the second. The influence *I* is a number ; thus, influence can decrease the absolute magnitude of the indirect effect but cannot increase it.

Influence is computed differently for cooperation curves and satisfaction curves; see Sections TBD and TBD for the details.

#### Neighborhood Proximities

Neighborhood proximity is defined by the 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 need not be symmetric; it is okay if . 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:

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 , and can then go on to define , , and so forth.

#### Here, Near, and Far Factors

The influence *I* determines the maximum indirect effect a direct effect may have on another curve. 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

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.

#### Magnitude of Indirect Effects

Given the terms defined in the preceding sections, we can now define the indirect effects resulting from a satisfaction or cooperation input to URAM. A direct effect on curve *i* with magnitude will have an indirect effect on curve *j* when the indirect magnitude is non-zero. Let

where

= The magnitude of the indirect effect on curve *j*

= The magnitude of the direct effect on curve *i*

= The influence of curve *i* on curve *j*.

*s*, *p*, *q* = The hear, near, and far factors for this input.

*m*, *n* = The relevant neighborhoods, where *j* is associated with *m* and *i* with *n*.

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

### Cooperation Levels

URAM 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*. URAM does not model collaboration.[[11]](#footnote-11)

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 . 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:[[12]](#footnote-12)

|  |  |  |  |
| --- | --- | --- | --- |
| SYMBOL | MEANING | VALUE | RANGE |
| AC | Always Cooperative | 100.0 |  |
| VC | Very Cooperative | 90.0 |  |
| C | Cooperative | 70.0 |  |
| MC | Marginally Cooperative | 50.0 |  |
| U | Uncooperative | 30.0 |  |
| VU | Very Uncooperative | 10.0 |  |
| NC | Never Cooperative | 0.0 |  |

#### Type Parameters

The initial baseline and natural levels are defined by the client simulation. We assume that the baseline changes slowly, and that it regresses slowly back to the natural level. Thus, the *α*, *β*, and *γ* parameters default to the following:[[13]](#footnote-13)

The natural level of the curve is specified by the client.

#### Cooperation Influence

The spread of indirect effects, described in Section 5.2, depends on the influence of one cooperation curve on another, which is to say the influence of one pair of groups on another. This section explains how cooperation influence is computed for such a pair of curves.

TBD.

#### Neighborhood Cooperation

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

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 is 0.

## 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 , each RMF returns a multiplier *r*. This *r* is typically multiplied by the nominal magnitude of an attitude input to produce the actual input.

### 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:[[14]](#footnote-14) the practical range is more like , 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[[15]](#footnote-15) called , where . This is the *nominal relationship*, the relationship that the subject matter experts should keep in mind when writing attitude rules. When , 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*.

Specific Relationship Multiplier Functions

Mars defines the following RMFs:

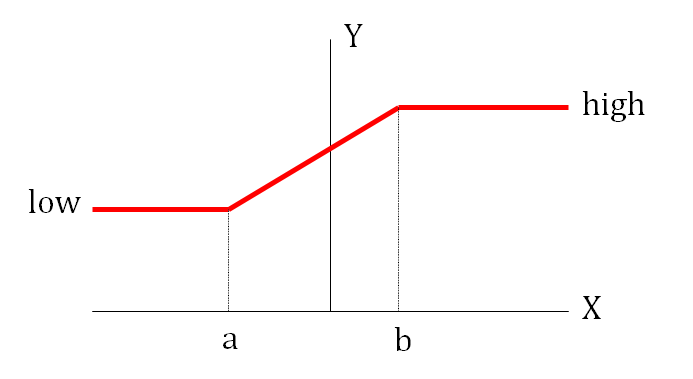
|  |  |
| --- | --- |
| **Constant:** The "Constant" function returns a constant 1.0, regardless of *R*. It is provided for use in generic code that takes the name of the RMF as an input, to remove the effect of group relationships. |  |
| **Linear:** The "Linear" function returns a value directly proportional 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. |  |
| **Quad:** The "Quad" function is similar in effect to the "Linear" function; however the resulting effect will be weaker than "Linear" for and stronger than "Linear" for . It is computed as follows: |  |
| **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. |  |
| **Friends More:** The "Friends More" function has a positive effect on both friends and enemies, but friends are affected much more strongly than enemies. |  |
| **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. |  |
| **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. |  |

## Miscellaneous Models and Algorithms

This section documents models and algorithms of general use.

### 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,

### 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, , determines that probability. The probability that *n* events will occur during an interval of length *t* is given by:

If the interval of time is always the same, the formula can be used to restate this formula for the probabilities recursively as follows:

### 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

## 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

1. Note that these proximities are social, not geographic—proximities are input to URAM, and are not computed from the geometry of the neighborhoods or the distance from one neighborhood to another. [↑](#footnote-ref-1)
2. This scale is implemented by the qposition type in simtypes(n). [↑](#footnote-ref-2)
3. This scale is implemented by the qemphasis type in simtypes(n). [↑](#footnote-ref-3)
4. We had originally thought that if were zero for some some topic *i,* then that topic should have no effect on . Later we realized that group *f* would still care about the groups around them, even if they did not care about *i.*  [↑](#footnote-ref-4)
5. See the memo “Mars Affinity Model”, by William H. Duquette (whd12\_002), 1 February 2012, for the derivation of these special cases. [↑](#footnote-ref-5)
6. Adjustments of this sort were originally implemented in JRAM to support course corrections during training with JNEM. Athena v4 supports adjustments, but the capability is only intended as an aid to testing. [↑](#footnote-ref-6)
7. It is expected that the client software will assign names to the causes; but for purposes of applying effects to curves it is more efficient to use integer IDs. [↑](#footnote-ref-7)
8. Remember that the adjustments were applied to the baseline between time advances. [↑](#footnote-ref-8)
9. There shouldn’t be any, but if there are then has already been changed by the client simulation and we must accommodate that. [↑](#footnote-ref-9)
10. Inputs for horizontal and vertical relationships do not have indirect effects in URAM. [↑](#footnote-ref-10)
11. Ambassador Terry McNamara has pointed out that "collaboration" is a loaded word, and that a more neutral term is preferable. [↑](#footnote-ref-11)
12. This scale is defined by the qcooperation(n) data type. [↑](#footnote-ref-12)
13. See uram.factors.COOP in the model parameter database. [↑](#footnote-ref-13)
14. 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. [↑](#footnote-ref-14)
15. Model parameter: rmf.nominalRelationship [↑](#footnote-ref-15)