SCAMP User Manual
Chapter 3: Experimental Files

H. Van Dyke Parunak and Jason Greanya

Chapter 1 introduced the distinction between the modeler and the experimenter, and Chapter 2 reviewed the files that make up a complete model. This chapter takes the perspective of someone experimenting with an existing model. This person might be a modeler who wants to develop the model incrementally, or someone who is exploring the behavior of an existing model.

The most direct way to experiment with a model is to edit **model.xlsx** and run SCAMP again. SCAMP will detect if the model.xlsx file is newer than the **model.graphML.xml** file that it builds from it, and will rebuild the model.graphML.xml file automatically in a few seconds. (Similarly, if you modify the geospatial map, **map.ora**, and start SCAMP, SCAMP will recompile the gradient database, but this process takes several hours.)

This chapter documents some other mechanisms for modifying the model that do not require rebuilding SCAMP's internal databases, and that also give you access to functionality that is not available through the files documented in Chapter 2.

- We begin with the Repast **parameters.xml** file, which defines parameters mostly concerned with the technical execution of the model. You can modify these parameters directly through the Repast graphical user interface. In our work so far, these parameters are set by SCAMP engineers, and modelers have not modified them. But some of them will be useful for experimentation.
- The **agents.csv** file allows you to overwrite any of the values from the fields in the groups tab that define the number and behavior of individual agents.
- Three files allow you to modify the events or geospatial regions that are included in a given run: **excluded.csv**, **agencyRanges.csv,** and **geoagency.csv**.
- **batch.csv** file makes it easy to execute multiple runs with different parameters.

1 parameters.xml

Many parameters in SCAMP are set using the Repast parameter mechanism, which allows them to be manipulated in a panel in the Repast interface or swept in Repast's internal batch mechanism. These parameters vary along two dimensions: the functional area that it helps control (model structure, agents, digital pheromones, determinism, execution control, or logging), and the perspective to which it belongs (Events, Goals, Geospatial movement, Social networking). Table 1 is an alphabetical list of all parameters, noting for each the function and perspective to which it applies. Then we discuss the parameters, grouped first by function and then by perspective. Bold-face parameters are those that experimenters are likely to find most useful. In the following list, the "Display Name" is the name by which the parameter is identified in the Parameters panel of the Repast interface, while "Parameter" is the name used in the program, and cached in the parameters.json file included in each run's log.

Table 1: List of SCAMP Parameters managed by Repast

Display Name	Parameter	Function	Perspective
Affiliation on All Features?	affiliateOnAllFeatures	Agents	Social
Avatar Determinism	**avatarDeterminism**	Determinism	Event
Avoid GeoSpace Cycles	**avoidGeoCycles**	Agents	Geospace
Base Geo Transit Time	baseGeoTime	Structure	Geospace
Batch run number	runNum	Execution	All
Default Random Seed	randomSeed	Execution	All
Display Buffer	displayB	Execution	All
Display Height	displayY	Execution	All
Display Width	displayX	Execution	All
Enable/Prevent Base	constructBase	Structure	Event
Enable/Prevent Scale	constructScale	Structure	Event
Enhance Base	enhanceBase	Structure	Event
Enhance Range	enhanceRange	Structure	Event
Files: Stickiness?	useStickiness	Agents	Event
Files: Symbolic features and preferences?	useSymbols	Execution	All
Files: Use HGNs?	**useHGNs**	Execution	All
GeoSpace deposit	**geoDep**	Pheromone	Geospatial
GeoSpace evaporation rate	**geoEvap**	Pheromone	Geospatial
GeoSpace Goal Determinism	**geoGoalDeterminism**	Determinism	Geospace
GeoSpace Goal Multiplier	geoGoalMult	Agents	Geospace
GeoSpace Goal Preference	geoGoalPref	Agents	Geospace
GeoSpace Lost Ratio	geoSpaceLostRatio	Agents	Geospace
GeoSpace Presence Preference	geoPresPref	Agents	Geospace
GeoSpace Terrain Preference	geoTerrPref	Agents	Geospace
Ghost Determinism	**determinism**	Determinism	Event
Ghost queue timer	queueTimer	Agents	Event
Group Change Dep	**groupChangeDep**	Pheromone	Social
Group Change Evap	**groupChangeEvap**	Pheromone	Social
Inhibit Base	inhibitBase	Structure	Event
Log Affiliations?	**logAffiliations**	Logging	Event
Log agents for queries?	**logAgentHistory**	Logging	Event
Log agents geospace travels	**logAgentTrack**	Logging	Geospace
Log Avatar Meetings?	**logMeetings**	Logging	Social
Log Directory	**logDir**	Execution	All
Log entropy?	**logEntropy**	Logging	Event
Log Events?	**logEvents**	Logging	Event
Log Features?	**logFeatures**	Logging	Event
Log full agent reports?	**logAgents**	Logging	Event
Log ghosts?	**logGhostQ**	Logging	Event
Log Graph Analysis?	**logGraphAnalysis**	Logging	Event
Log HGN satisfaction?	**logSatisfaction**	Logging	Goal
Log influences?	**logInfluences**	Logging	Event
Log Preferences?	**logPreferences**	Logging	Event
Log Roulettes?	**logRoulettes**	Logging	Event
Logs: Reset Between Runs?	**reset**	Logging	Event

Display Name	Parameter	Function	Perspective
Max Ghost Ticks	maxGhostTicks	Agents	Event
Model Directory	**fileRoot**	Execution	All
Node deposit	**nodeDep**	Pheromone	Event
Node evaporation rate	**nodeEvap**	Pheromone	Event
Realized Network Count Weight	interCountWeight	Pheromone	Social
Realized Network Evap	interEvap	Pheromone	Social
Run Duration	**runDur**	Execution	All
Run Length	**runLen**	Execution	All
Save snapshot for restart	snapshot	Execution	All
Sigmoid slope parameter	sigParam	Pheromone	Event
Sigmoid type	sigType	Pheromone	Event
Sync Restarting Agents	syncRestarts	Agents	Event
Target Spatial Nodes	targetSpatialNodes	Structure	Geospace
Update model.graphML.xml	updateXML	Execution	All
Urgency Baseline	urgencyBaseline	Agents	Social
Use file-based zips?	useZipFile	Structure	Goal
Use geospace?	**useGeoSpace**	Execution	All
Use Hard Influences	useHardInfluences	Structure	Event
Use Influences	useInfluences	Structure	Event
Use Multithreading	useThreading	Execution	All
Use Social Networking	**useSocial**	Execution	All
Use Soft Influences	useSoftInfluences	Structure	Event
Variable Pheromone Deposits	variableDeposits	Pheromone	Event
Wellbeing Active?	wellbeing	Agents	Event
Wellbeing balance increment	wbi	Agents	Event
Wellbeing preference increment	wpi	Agents	Event

1.1 Model Structure

The heart of a SCAMP model is a set of graphical structures that characterize each of its perspectives. Some of the parameters govern the interpretation of these structures.

1.1.1 Event Perspective

Influence parameters (Table 2) control the behavior of influence edges in the CEG. In particular, most of these parameters compute the factors that govern the influence of source pheromones on the likelihood that the target event will be available for agents to access over agency edges.

Table 2: Influence Parameters

Display Name	Parameter	Type	Default	Explanation
Enable/Prevent Base	constructBase	double	0.99	Base of the enable and prevent factors; raised to a power dependent on pheromone level on source event
Enable/Prevent Scale	constructScale	double	500.0	Scale factor for the exponent in computing enable and prevent factors
Enhance Base	enhanceBase	double	0.99	Base of the enhance factor; raised to a power dependent on pheromone level on source event
Enhance Range	enhanceRange	double	1.0	How much influence source pheromones have on enhance factor, in [0, 1].

Inhibit Base	inhibitBase	double	0.5	Base of the inhibit factor; raised to a power dependent on pheromone level on source event
Use Hard Influences	useHardInfluences	boolean	true	Determines whether hard influences modulate the probability of adding an event to the roulette.
Use Influences	useInfluences	boolean	true	If false, overrides both useHardInfluences and useSoftInfluences; useful in batch configurations
Use Soft Influences	useSoftInfluences	boolean	true	Determines whether soft influence edges in the CEG (enhance, inhibit) modulate the size of roulette segments.

1.1.2 Goal Perspective

Display Name	Parameter	Type	Default	Explanation
Use file-based zips?	useZipFile	boolean	true	Load zips from separate set of columns in the events tab of model.xslx, rather than having HGNs and the CEG all in a single graph

1.1.3 Geospatial Perspective

Display Name	Parameter	Type	Default	Explanation
Base Geo Transit Time	baseGeoTime	int	1	Defines the default transit time for geospatial nodes, which may be modulated by terrain type and group
Target Spatial Nodes	targetSpatialNodes	int	400	Approximate number of geospatial nodes desired (exact number depends on alignment of hex tiles)

1.1.4 Social Perspective

The groups are defined by the groups tab in the workbook, but their structure has no Repast parameters.

1.2 Agents

1.2.1 Event Perspective

Display Name	Parameter	Type	Default	Explanation
Files: Stickiness?	useStickiness	boolean	false	Use the stickiness information in the events tab of model.xslx to decide whether to include an agent's current event in its roulette (thus allowing it to repeat that event immediately)
Ghost queue timer	queueTimer	int	10	How many ticks a ghost is allowed to wait in the queue of a ThenManyIn relation for other required agents to arrive
Max Ghost Ticks	maxGhostTicks	int	10	Maximum number of ticks a ghost is allowed to run
Sync Restarting Agents	syncRestarts	int	false	An agent may need to restart for two reasons: it gets lost in geospace (in which case it returns to the last event where it had a choice of successors), or it reaches a node with no successor (in which case it returns to start). If syncRestarts is true, no agent can restart more times than the other avatars..

Wellbeing Active?	wellbeing	boolean	true	Whether or not an agent's wellbeing is modified by the wellbeing features of each event
Wellbeing balance increment	wbi	double	0.5	Wellbeing increment: multiplier applied to an event's wellbeing features to derive the total modification to the agent's wellbeing
Wellbeing preference increment	wpi	double	0.5	Wellbeing preference increment: if wellbeing preferences are modified, this adjusts the influence that WB has on the preferences. 0 → they don't change.

1.2.2 Goal Perspective

There are no Repast parameters directly concerning agents in the Goal perspective.

1.2.3 Geospatial Perspective

Display Name	Parameter	Type	Default	Explanation
Avoid GeoSpace Cycles	**avoidGeoCycles**	boolean	true	If true, reduces the probability of revisiting a geonode proportionally to how often this agent has visited it before on this trip. Turning this on (the default) makes it less likely that agents will get lost in geospace.
GeoSpace Lost Ratio	geoSpaceLostRatio	double	0.125	Proportion of total hex nodes that an agent must visit before deciding it is lost in geospace
GeoSpace Goal Multiplier	geoGoalMult	double	5.0	Multiplier for gradients to geo destinations to enhance their impact on ghost decisions
GeoSpace Goal Preference	geoGoalPref	double	5.0	Value of agent's urgency preferences for geospatial destinations when in geospace
Geo Persistence	geoPersistence	Double	0.75	How persistently an agent will continue to attempt to reach its goal if it gets lost and aborts its effort. A value of 1 means the agent will repeat the effort until it succeeds or the run ends. A 0 means that after a single failure, it will return to START.
GeoSpace Presence Preference	geoPresPref	double	1.0	Multiplier for agent's presence preferences when in geospace
GeoSpace Terrain Preference	geoTerrPref	double	1.0	Value of agent's preferences for static features in geospace, where these features reflect terrain difficulty

1.2.4 Social Perspective

Display Name	Parameter	Type	Default	Explanation
Affiliation on All Features?	affiliateOnAllFeatures	boolean	true	Agents decide whether to affiliate with groups other than their own based on the dot product of their preferences with the base preferences of other groups. Should they use preferences over all features, or only over wellbeing features?
Urgency Baseline	urgencyBaseline	double	0.3	Added to baseline (normally 0) of urgency pheromone for a group for events that support transitions to that group

1.3 Digital Pheromones

Agents interact with one another by augmenting and sensing variables in the locations in their environments that they visit. Because these variables are inspired by insect pheromones, we call them "digital pheromones." Each such interaction is governed by two parameters: a *deposit* parameter that specifies the amount by which the agent augments the variable, and an *evaporation* parameter that specifies how much the variable decays with each domain day. In each case, the evaporation parameter is defined in terms of the proportion of the variable that is lost, so the evaporation itself consists of multiplying by (1 – *evap*).

Digital pheromones saturate (usually at 1), so deposit is done through a sigmoid function. Deposits have most effect when the variable is at half value. The closer the variable is to the upper limit, the less impact each new deposit has.

SCAMP has two kinds of agents: avatars, which represent domain entities, and ghosts, which avatars deploy to think into the future. An important difference between the two is in how they use pheromones.

- Ghosts deposit presence pheromones on the nodes they visit. Avatars do not deposit any pheromones.
- Ghosts have preferences over all of the features that events and geospatial locations present, including not only presence pheromones, but urgency pheromones and wellbeing features. Avatars attend only to the presence pheromones of groups with which they are affiliated.

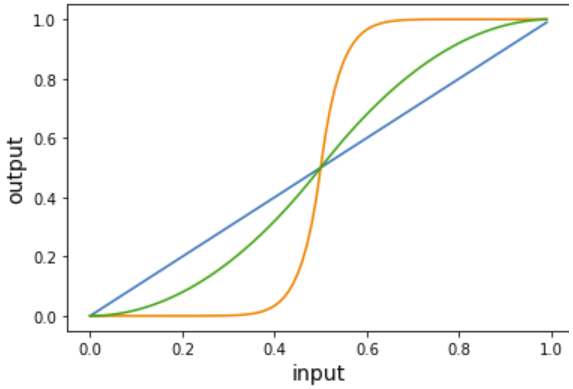

Figure 1: Variation of quadratic sigmoid for sigParam = -1 (blue), 0 (green), 10 (orange)

In other words, ghosts do all the reasoning about choices among options. Avatars simply follow the presence pheromones laid down by their ghosts.

1.3.1 Event Perspective

In the CEG, agents deposit and sense pheromones on event nodes to record their participation.

Display Name	Parameter	Type	Default	Explanation
Node deposit	**nodeDep**	double	0.2	Amount deposited by a ghost in its group's presence feature on CEG events that it visits. Increasing this increases the contribution that each ghost makes to the pheromone concentration on an event.
Node evaporation rate	**nodeEvap**	double	0.01	Amount evaporated from presence pheromones on CEG events each domain day. The presence pheromone is multiplied by (1 – nodeEvap) each day. The higher this value, the more quickly events forget the presence of ghosts.
Sigmoid slope parameter	sigParam	double	1.0	Controls slope of sigmoid deposit function, varying from a straight line to a step function. The values used depend on the sigmoid type. sigParam has no effect on sigType

				lin(ear). For seg(mented) and cau(chy), it is the slope of the sigmoid at the halfway point, but as implemented, these methods do not currently consult it. For qua(dratic), -1 gives a step function, 0 gives a parabola, and positive values move toward a step function (Figure 1), but it may be frozen in code once tuned.
Sigmoid type	sigType	string	qua	Selects function used for limiting pheromone deposits. The options are lin(ear) (linear addition, truncated at upper limit), qua(dratic) (based on parabola, $y = ax^2$, raised to the power p), cau(chy) (original function based on inverted Cauchy distribution), and seg(mented) (an approximation to the Cauchy function for efficiency purposes). The quadratic with parameter -1 and linear give the same result, but the linear is more efficient.
Variable Pheromone Deposits	variableDeposits	double	1.0	If 0, all ghosts make same size deposit, regardless of the quality of the solution they find. If 1, deposits are proportional to the qualify of the solution. Quality is the sum of the exponentiated dot products of the ghost's preferences and the feature vectors of the chosen events. Intermediate values reduce the impact of quality, for experimental purposes. In production, this value should be left at 1.0.

1.3.2 Goal Perspective

There are no Repast parameters for pheromones in the Goal perspective

1.3.3 Geospatial Perspective

In the geospatial perspective, agents deposit pheromones on the nodes that tile the map to record their presence.

Display Name	Parameter	Type	Default	Explanation
GeoSpace deposit	**geoDep**	double	0.2	Amount deposited by a ghost in its group's presence feature on geospatial nodes that it visits.
GeoSpace evaporation rate	**geoEvap**	double	0.05	Amount evaporated from presence pheromones on geospatial nodes each domain day. The presence pheromone is multiplied by (1 – geoEvap) each day.

1.3.4 Social Perspective

Display Name	Parameter	Type	Default	Explanation
Group Change Dep	**groupChangeDep**	double	0.6	How much an agent is likely to change groups, given the difference between its perceived and its actual probability of interaction with agents of another group. Increasing this makes the agent more susceptible to each interaction with an agent from another group.
Group Change Evap	**groupChangeEvap**	double	0.7	Amount evaporated from an agent's group change preference each domain day. The preference is multiplied (1 – groupChangeEvap) each day.

Display Name	Parameter	Type	Default	Explanation
Realized Network Count Weight	interCountWeight	double	0.5	The interaction weight between two agents is a weighted average of the normalized number of times they've interacted and their normalized interaction time. This parameter gives the weight of the count; the weight of the time is 1 – interCountWeight.
Realized Network Evap	interEvap	double	0.05	Interaction weight evaporates by this fraction each day of domain time, by multiplying it by 1 – interEvap.

1.4 Determinism

1.4.1 Event Perspective

Determinism parameters adjust the degree of stochasticity in the system. Agents select their next move by spinning a roulette weighted proportionately to the dot product of their preferences with the features of the options they are considering. We raise each of the roulette segments to a fixed power, the determinism, then normalize. If determinism = 0, all segments are of equal width, and the system is not constrained by the preference-feature dynamics at all. If determinism is high (say, 10), the smaller options become vanishingly small compared with the largest one, and the system behaves deterministically. Different determinism values impact different choices.

Recall that

- Avatars attend only to presence pheromones, allowing ghosts to do all the reasoning about wellbeing and urgency dimensions
- The urgency dimensions are used differently in event space and geo space. In event space, they reflect the state of the HGNs for the different groups. In geo space, they reflect the gradient to the geospatial goals for each group.

Display Name	Parameter	Type	Default	Explanation
Avatar Determinism	**avatarDeterminism**	double	5.0	Determinism for decisions made by avatar agents. Increase (decrease) this parameter to have avatars follow the recommendations of their ghosts more (less) closely.
Ghost Determinism	**determinism**	double	5.0	Determinism for decisions made by ghost agents. Increase (decrease) this parameter to have ghosts explore fewer (more) options.

1.4.2 Goal Perspective

There are no Repast parameters governing determinism in the Goal perspective.

1.4.3 Geospatial Perspective

Geospace is less structured than event space (offering an agent six choices at each step, rather than typically two or three as in the CEG), so we have found it necessary to increase the determinism for agents moving in geospace.

Display Name	Parameter	Type	Default	Explanation
GeoSpace Goal Determinism	geoGoalDeterminism	int	10	Additional determinism applied by ghost agents to urgency pheromones in geospace

1.4.4 Social Perspective

There are no Repast parameters governing determinism in the Social perspective.

1.5 Execution Control

The Execution parameter are shared by all perspectives, and provide high-level control for the overall run.

Display Name	Parameter	Type	Default	Explanation
Batch run number	runNum	int	1	A number that can be incremented in batch runs to distinguish the results of the different runs from one another.
Default Random Seed	randomSeed	int	__NULL__	The default value results in the assigning of a new random seed to each run, based on the computer's clock. To ensure reproducibility, the parameter can be set to a fixed value.
Display Buffer	displayB	double	5.0	Display buffer: space to leave at each side of display before plotting objects
Display Height	displayY	double	60.0	Height of displays
Display Width	displayX	double	90.0	Width of displays
Files: Symbolic features and preferences?	useSymbols	boolean	true	Use symbols (l, m, s, 0, S, M, L) instead of numbers to specify group preferences and event static features. This parameter is no longer used. The Excel spreadsheet can use either these symbols or numbers, and the current code will adjust.
Files: Use HGNs?	**useHGNs**	boolean	true	Activate the goal perspective. (Even with this parameter set to *true*, the use of HGNs for individual groups can be turned off individually in the **groups** tab of **model.xlsx** in the `UseHGN` column,
Log Directory	logDir	string	logs	Path to directory where all log files can be found. If a relative path (as here), the files are under the SCAMP directory, but users can specify an absolute path anywhere on their machine.
Model Directory	**fileRoot**	string	data/SCAMP.CEG0073	Path to directory where all configuration files can be found. If a relative path (as here), the files are under the SCAMP directory,

				but users can specify an absolute path anywhere on their machine.
Run Duration	**runDur**	int		Maximum length of a run in domain time. When domain time reaches this value, the run terminates.
Run Length	**runLen**	int		Maximum length of a run in ticks (the basic cycle of agent execution). When the current tick reaches this value, the run terminates. (But see discussion below in case of a resumed run.)
Save snapshots for restart	snapshots	boolean	true	Save initial and final snapshots, allowing run to be restarted These files can be very large, so turning this off can reduce storage requirements if restarting is not needed.
Update model.graphML.xml	updateXML	boolean	true	In batch processing, file times are changed, and turning this off can save run time
Use geospace?	**useGeoSpace**	boolean	true	Activate the geospatial perspective
Use Multithreading	useThreading	boolean	true	Turn on multithreading to reduce runtime on multi-core processors
Use Social Networking	**useSocial**	boolean	true	Activate the social perspective

Sometimes it is desirable to resume a previous run. snapshot.obj.gz, a log file, contains the entire state of the run. If you copy this file to data/<modelName> (the same directory as the model.graphML.xml file) and start the sim, the new run will pick up where the old one left off, with one exception: the tick counter will start over. In this case, the runLen parameter defines the number of *additional* ticks that the sim will run. For example, if the initial run was for 2000 ticks, then setting the runLen to 2000 for the subsequent run will run from 2001 to 4000. The UI will still show the tick as 1 to 2000, but all of the console output, logs, internal counts, etc. will show 2001 to 4000. The original runDur parameter still applies; if the original run terminated because the sim exceeded runDur, no further extension can be run.

For restarted runs, the **model.graphML.xml** file, **agents.csv**, etc. are ignored, but a subset of the parameters are still read and used: fileRoot, runLen, nodeDep, nodeEvap, geoDep, and geoEvap. This is a mixed blessing. The root directory and run length can be changed, but nodeDep, etc. can also be changed. So, if the user isn't careful with the parameters, basic workings of the sim can be altered because the parameters were inadvertently changed. On the plus side, if you want to change an evaporation value between tick 2000 and tick 2001, you can.

The log directory stays the same as the previous run, ignoring the logDir parameter, and new lines are appended to existing logs. The parameters.json file is updated with a "totalRunLength" parameter. So for the previous example, runLen would still be 2000, but totalRunLength would be 4000. Also, under an "old" parameter heading is the differences between the previous parameters and the current run. When the run completes, the snapshot in the log directory is overwritten with the new ending snapshot.

Setting the runLen parameter involves a balance between how long you can wait for results and how meaningful your results will be. A major difference between a simulation and the real world is that a stimulation starts from scratch, while the real world has been running for a long time. When you start any system up, it goes through transients that say more about the fact that it's starting de novo than about its intrinsic dynamics. Thus you need to run the system long enough to get past the start-up dynamics. But how long is long enough?

There are two ways to answer this question.

1. Do the logs show that the system has entered an attractor of some sort? The attractor could be chaotic, or even random, which could be hard to detect, but you can detect when it is a fixed point or a cyclic attractor.
2. Given the structure of SCAMP, you might want to look at how many agents get to the end of the CEG and start over. You may very well never visit some nodes, but if agent has made it through the entire CEG, you haven't really explored the impact of the ends of the various histories that the CEG can generate.

1.6 Logging

Though Repast provides a graphical interface, the size of our models is too large to be examined conveniently, so SCAMP usually runs in batch mode, producing a variety of logs for analysis and post-processing. Chapter 4 describes the contents of these logs in detail. Logging parameters allow the user to control which logs are recorded.

1.6.1 Event Perspective

Display Name	Parameter	Type	Default	Explanation
Log Affiliations?	logAffiliations	boolean	true	Produce affs.csv indicating each avatar's affiliation over time
Log agents for queries?	logAgentHistory	boolean	true	Produce agentHistory.csv indicating the history of events in which each avatar has participated (and, if geospace is active, agentTrack.csv recording the series of locations that each agent has visited)
Log entropy?	logEntropy	boolean	true	Produce entropyLog.csv, containing time series of entropy overall and by group, and a variety of other execution-related statistics
Log Events?	logEvents	boolean	true	Produce eventLog.csv, which records the total number of ghosts and avatars to visit each event in the CEG over the entire run.
Log Features?	logFeatures	boolean	true	Produce features.csv, recording the feature values of each event or geospatial hex at each domain time
Log full agent reports?	logAgents	boolean	false	Produce agentLog.csv, which largely overlaps with agentHistory.csv, but includes each agent's history
Log ghosts?	logGhostQ	boolean	false	Produce ghostLog.csv, which records the state and history of each ghost during a run—very voluminous
Log Graph Analysis?	logGraphAnalysis	boolean	false	Produce graphLog.txt, reporting structural problems in the CEG (e.g., dead ends, unreachable nodes, cycles). This option is useful in debugging a new CEG, but can be very time consuming, so if an experiment does not involve changing the structure of the CEG, it is best to turn it off.

Log influences?	logInfluences	boolean	true	Produce influences.csv, recording the kind and weight of all influences among CEG events over time
Log Preferences?	logPreferences	boolean	true	Produce fullPrefs.csv, which records each agent's preferences over all accessible events each time it faces a choice
Log Roulettes?	logRoulettes	boolean	true	Produce rouletteLog.csv, which records the raw roulette (inner product of each option, no exponentiation or determinism) each time an agent computes it.
Logs: Reset Between Runs?	reset	boolean	true	In batch mode, start new set of logs for each configuration running on an instance.

1.6.2 Goal Perspective

Display Name	Parameter	Type	Default	Explanation
Log HGN satisfaction?	logSatisfaction	boolean	true	Produce satLog.csv, recording the satisfaction at the root of each HGN over time

1.6.3 Geospatial Perspective

Display Name	Parameter	Type	Default	Explanation
Log agents geospace travels	**logAgentTrack**	boolean	true	Produce agentTrack .csv, which records the location of each agent over time.

This parameter can be overridden in two ways: if either useGeo or logAgentTrack is false, no agent track log will be generated.

1.6.4 Social Perspective

Display Name	Parameter	Type	Default	Explanation
Log Avatar Meetings?	logMeetings	boolean	true	Produce meetings .csv, which records the total number and duration of meetings for each pair of agents

2 agents.csv

This file defines the initial population of agents that exercise the model. It overwrites the information in the groups tab of model.xlsx, in the columns IndependentAgency, ReceivesAffiliation, MakesAffiliation, GhostsPerShift, ShiftsPerGen, MaxGhostSteps, NumGroupAvatars, NumIndividualAvatars, AffiliationThreshold, PreferenceVariation, and UseHGN. It does not need to include every group, every column, or every value in the groups and columns that it does contain, and missing values retain the values they have in model.xlsx, but it does need the id column, and column headers must be labeled the same way they are in model.xls. For example, consider an agents.csv with these contents:

id	NumIndividualAvatars	AffiliationThreshold	UseHGN
2	10	.5	
4	10		0

This file will change only groups 2 and 4. It will assign 10 individual avatars to each group, regardless of what model.xls says. It will change the affiliation threshold of agents in group 2 to 0.5, and will turn off the HGN for group 4. All other values remain as specified in model.xls.

3 Dynamic modification of agency

A central concept in SCAMP is that different groups "have agency" for different events or geospatial locations, and agents can only participate in an event (or visit a location) if a group with which they are affiliated has agency for that event or location. Causal influence among events for which disjoint sets of groups have agency is handled in four ways: influence edges in the CEG, different events being zipped to the same HGN, agents participating in different geospatial events but encountering one another in geospace, and group change actions (specified in the groupChanges tab of model.xlsx). But agents on an event or location can only choose an adjacent event or location if they have agency for that event or location.

The events tab in the workbook includes an Ag_ column for each group, defining agency for events. By default, all agents have agency for all locations in geospace. The files in this section allow you to change the agency for some events or locations, either throughout the entire run or for certain period of time. In other words, you can prune away portions of event space or geospace, making them inaccessible to agents of specified groups. This section discusses these files and the issue of precedence among different ways of defining agency.

3.1 excluded.csv

Table 3: Example excluded.csv

id
3
24

This file allows one to remove some events from the CEG for a single run. SCAMP pays attention only to the column labeled id, which should contain the event number (as in the events tab of **model.xlsx**), but you can include the event name in another column for your own convenience. **excluded.csv** cannot directly manipulate geospatial locations. For example, the example in Table 3 effectively removes events 3 and 24, with all their links to the rest of the model, from the CEG. It applies to all agents, without respect to group.

3.2 agencyRange.csv

This file allows agency to be adjusted on a group-by-group basis, and to change during the run. It has five columns:

- EventID: this value can take one of three forms: an event number from the id column of the events tab, an event ID of the form Ennn where nnn is the event id, or a region ID of the form Rnnn from the regions tab
- EventName: the textual name of the event or region (for experimenter convenience, ignored by the code)
- Group: a group id from the groups tab
- StartInclusive: the first domain time at which the agents of the specified group should be given agency for the specified event.
- EndInclusive: the last domain time at which the agents of the specified group should be given agency for the specified event.

Table 4 says that groups 3 and 4 will have agency on event 5 for domain times 1 through 100, inclusive, overriding all other agency information for those groups on event 5 in the events tab of model.xls.

Table 4: Example of agencyRange.csv

EventID	EventName	Group	StartInclusive	EndInclusive
5	Event5Name	3	1	100
5	Event5Name	4	1	100

Table 5 excludes all agents from a geospatial region. These lines allows all groups into R021 only for domain times 0 to 0, inclusive, which actually excludes them, since there's no domain time 0.

Table 5: Geospatial example of agencyRange.csv

EventID	EventName	Group	StartInclusive	EndInclusive
R021	South Peggytown	0	0	0
R021	South Peggytown	1	0	0
R021	South Peggytown	2	0	0
R021	South Peggytown	3	0	0
R021	South Peggytown	4	0	0
R021	South Peggytown	5	0	0
R021	South Peggytown	6	0	0

Notes:

- Each line adds agency for a single group, so multiple lines are necessary if multiple groups need agency on a given event.
- agencyRange.csv does not affect events that it does not mention at all.
- Multiple rows can be defined for an event-group pair so that it can be active over multiple ranges, say from 0 to 1000 and 2001 to 3000:

3.3 geoagency.csv

For the geospace model, all hexes by default have agency for all groups. The agencyRange file allows the user to change agency by region, but sometimes one may want to change the agency of individual hexes in geospace (for example, to define a barrier that does not have a region name). This file provides that functionality. Its columns are:

- x: the x hex coordinate (from left to right). In our current configuration, x is in [0, 23], and 0 is the column of half-hexes at the left.
- y: the y hex coordinate (from top to bottom). In our current configuration, y is in [0, 21].
- Group: the group id from the groups tab identifying which groups' agency is controlled by the line
- Start: the first domain time at which the group will have agency for the hex.
- End: the last domain time at which the group will have agency for the hex.

As with agencyRange, multiple rows in georange.csv can define multiple ranges for the same hex-group pair. Full hex exclusion would also follow the same pattern as above, requiring a separate line for each group.

Hex coordinates are only valid if the image files stay the same size and the "targetSpatialNodes" Repast parameter stays the same. If either of those items changes, the hexes on the map may change size and/or number. Currently, targetSpatialNodes is 400, which yields a hex grid about 20 by 20, but if we increased that to 1000, we'd end up with a grid about 32 by 32 grid, so the x,y grid coordinates defined in geoagency.csv would no longer be valid.

3.4 Precedence among agency controls

In all, there are four ways to specify which groups have agency for which events or regions. The order of this list reflects the precedence among them: a file later in the list overrides agency constraints imposed by earlier files.

- By default, all groups have agency for all geospatial hexes.
- The Ag_ columns in the events tab in the workbook defines agency for CEG events. Each "1" entry gives agency for that column's group for all time. For example, if the E33 row has a 1 in the AgGov column, government agents have agency for E33 throughout the entire simulation run.
- excluded.csv turns off the agency for all groups for the listed events, for the entire simulation.
- geoagency.csv allows the agency for specified individual hexes to be restricted over specified periods of time.
- agencyRange.csv allows the agency for specified events and regions to be restricted over specified periods of time.

4 batch.csv

Repast allows the definition of batch runs that sweep any of the internal Repast parameters, but sometimes we want to vary groups of parameters concurrently, or redefine some of the SCAMP configuration files, in ways that Repast does not support. The batch.csv file supports this capability, and is active if it is present in the data directory *and* the Repast parameter file includes a runNum parameter.

batch.csv has four columns: start, end, name, and value.

- start and end are the runNum (both inclusive) where that row is in effect.
- The name and value columns are either the name of the parameter to be changed and its new value or the name of a file to be replaced and the file to replace it. When the name is a filename, it needs to start with the current fileRoot.

Table 6 shows a simple example. The first line says that the "determinism" parameter will be set to 1.0 when runNum is 1.

Table 6: Example of batch.csv

start	end	name	value
1	1	determinism	1.0
1	2	SCAMP.CEG0019.excluded.csv	data/batch.excluded.csv
2	2	determinism	5.0

The second line says that the "SCAMP.CEG0019.excluded.csv" will be replaced with the file at "data/batch.excluded.csv" when runNum is 1 and 2.

The third line sets the "determinism" parameter to 5.0 when runNum is 2.

There are a few caveats:

- Parameters that are not recognized just print out an error but don't do anything to stop the process. So a mistake may appear hours into a run without anyone noticing.
- The batch mechanism does not backup the file being replaced, so if it needs to be swapped back, there needs to be an available copy; it won't "default" back to what it was originally.

- Similarly, after a file swap, the last file swapped into that file name will persist after the batch run.
- The file name has to start with whatever the fileRoot parameter is and needs to be in the data directory.
- The file value needs a more complete place. It doesn't assume that it's in the data directory.
- Any parameter changes are only valid for that run and aren't stored anywhere longterm.
- All parameters that need to be changed have to be writeable

5 Resuming a Prior Run

Sometimes analysis of data from a run shows that it would be useful to have let the run continue longer. If the run involves many agents and includes geospatial activity, redoing the entire run can be very time-consuming. As long as your original runDur parameter is high enough to include both the original and the extended run, you can continue a previous run where it left off.

The logs for each run include a file entitled snapshot.obj.gz. This file contains the entire state of the simulation at the point that it stopped. If the model directory (e.g., …/SCAMP_Repast/SCAMP_C/data/SCAMP.CEG0073/model) contains this file when the model starts, the model will pick up where it left off.

A few details require attention.

1. The runLen parameter will be interpreted as the length of the *additional* run. For example, if the initial run was for 2000 ticks, then setting the runLen to 2000 for the subsequent run will run from 2001 to 4000. The user interface will still show the tick as 1 to 2000, but all of the console output, logs, internal counts, etc. are 2001 to 4000.
2. For restarted runs, the model.graphML.xml file, agents.csv, etc. are ignored, but subset of the parameters are still read and used: fileRoot, runLen, nodeDep, nodeEvap, geoDep, and geoEvap. This is a mixed blessing. The root directory and run length can be changed, but nodeDep, etc. can also be changed. So, if you inadvertently change the parameters, the basic workings of the sim can be altered. Each run captures all of its parameters in the parameters.json log file, and you should check to be sure that your parameters are consistent.
3. The log directory stays the same as the previous run, ignoring the logDir parameter, and new lines are appended to existing logs. The parameters.json file is updated with a "totalRunLength" parameter. So for the previous example, runLen would still be 2000, but totalRunLength would be 4000. Also, under an "old" parameter heading is the differences between the previous parameters and the current run. When the run completes, the snapshot in the log directory is overwritten with the new ending snapshot.
4. One consequence of the parameter handling is that the initial runDur needs to be high enough for subsequent runs to work.

Acknowledgements

The development of SCAMP was funded by the Defense Advanced Research Projects Agency (DARPA), under Cooperative Agreement HR00111820003. The content of this paper does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.