Documentation of PAR database (pardb)

October 2008

Jan Allbeck

This database consists of 39 largely unpopulated tables. The tables can be divided into 3 groups: action tables, object tables, and tables for our work on crowd simulations. There is one table that links them, obj_act. The original database was designed by Melina Schweizer in 2003 and has since been refined. In this document, we will describe the tables and their entries. See *PARdb.vsd* for a diagram of the database.

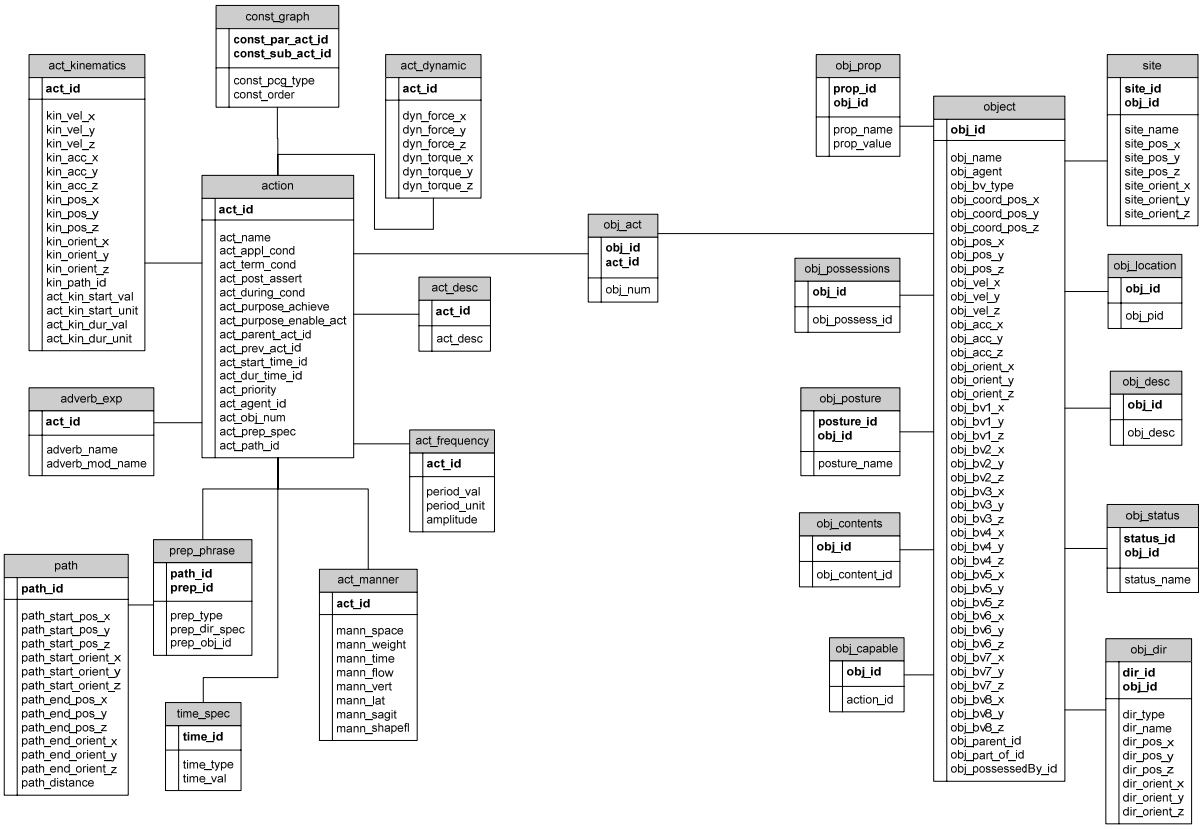

Figure 1: An outdated diagram of the database structure as of the writing of this document.

Table of Contents

1. Actionary

The following tables are mainly related to action PARs and are meant to store both uPARs and iPARs.

a. action

This table is the main table related to the action part of PARs. It is linked to other tables to save on storage costs.

Column name	Data type	Description
act_id	SMALLINT	The action id that is used to link this PAR to other tables including action properties and objects. [PRIMARY KEY]
act_name	VARCHAR	The name of the action should be similar to a verb.
act_app_cond	VARCHAR	The applicability conditions field could be a Python expression but is more likely a Python script filename. If the applicability conditions evaluate to false, the action will not be performed.
act_term_cond	VARCHAR	The termination conditions field could be a Python expression but is more likely a Python script filename. Termination conditions indicate when the action has been completed. It may be based on the duration of the action or the state of the world.
act_post_assert	VARCHAR	The post assertions field could be a Python expression but is more likely a Python script filename. The post assertions are executed after the action has terminated to update the state of the world.
act_during_cond	VARCHAR	The during conditions field could be a Python expression but is more likely a Python script filename. During conditions need to be held true through out the action execution. The action should fail if they become false.
act_purpose_achieve	VARCHAR	This field indicates the state of the environment that is trying to be achieved by performing the action. This field could be a Python expression or a Python script filename.
act_purpose_enable_act	SMALLINT	This field indicates which action this action is trying to enable. Its action id will be in this field.
act_parent_act_id	SMALLINT	This field holds the action's parent's id. This is used mainly for inheritance.
act_prev_act_id	SMALLINT	The previous action field holds the id of the action that was/is being executed before this action.
act_start_time_id	SMALLINT	This field holds an id that is referenced in the time_spec table. It refers to the start time of the action.
act_dur_time_id	SMALLINT	The duration time field holds an id that is referenced in

		the time_spec table and refers to the duration of the action.
act_priority	TINYINT	The priority level of the action.
act_agent_id	SMALLINT	The agent id corresponds to an object id in the database that should also be an agent process. This is the agent that is to perform the action.
act_obj_num	TINYINT	The object number is the number of objects that are participants in this action.
act_prep_spec	VARCHAR	The preparatory specifications are condition action pairs. The field could be a Python expression but is more likely a Python script filename.
act_path_id	SMALLINT	The path id references path information in the prepositional phrase and path tables. Path information can be found in these tables.
act_next_act_id	SMALLINT	The id of the action to follow this action.
act_exec_steps	VARCHAR	The executions steps (sub-steps) for this action. Actually a path and python file name that contains them.
act_group	TINYINT	A flag to indicate whether the iPAR is for an individual agent or a group of agents. If it is an individual, act_agent_id will contain the id of an agent (object). If it is a group, act_agent_id will contain the id of a group (from the groups table)
act_location	SMALLINT	The location where this action should take place. It contains an object id. This field is not required for an action, but if an object type is specified and not an object instance, then this location can be used to ground the action into a space.
act_site_type_id	SMALLINT	The id that indicates the type of site that should be used to properly locate the agent relative to the object participant of the action. For example, the position and orientation a character should stand in to clean a chair vs. to sit in a chair.

b. act_desc

A description of this action

Column name	**Data type**	**Description**
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
act_descript	VARCHAR	A small English description of the action or note about it.

c. act_dynamic

A description of the action dynamics including force and torque.

Column name	Data type	Description
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
dyn_force_x	FLOAT	The dynamic force of the action in the x direction.
dyn_force_y	FLOAT	The dynamic force of the action in the y direction.
dyn_force_z	FLOAT	The dynamic force of the action in the z direction.
dyn_torque_x	FLOAT	The torque of the action in x.
dyn_torque_y	FLOAT	The torque of the action in y.
dyn_torque_z	FLOAT	The torque of the action in z.

d. act_frequency

A description of the frequency of this action.

Column name	Data type	Description
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
period_val	FLOAT	The value for the period of the frequency.
period_unit	VARCHAR	The units of the period (time).
amplitude	FLOAT	The amplitude of the frequency.

e. act_kinematics

A description of the kinematics of the action.

Column name	Data type	Description
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
kin_vel_x	FLOAT	The velocity of the movement in the x direction.
kin_vel_y	FLOAT	The velocity of the movement in the y direction.
kin_vel_z	FLOAT	The velocity of the movement in the z direction.
kin_acc_x	FLOAT	The acceleration of the movement in the x direction.
kin_acc_y	FLOAT	The acceleration of the movement in the y direction.
kin_acc_z	FLOAT	The acceleration of the movement in the z direction.
kin_pos_x	FLOAT	The position in x.
kin_pos_y	FLOAT	The position in y.
kin_pos_z	FLOAT	The position in z.
kin_orient_x	FLOAT	The orientation in x.
kin_orient_y	FLOAT	The orientation in y.
kin_orient_z	FLOAT	The orientation in z.
kin_path_id	SMALLINT	A path id that corresponds to a path in the path table.

act_kin_start_val	INTEGER	The start time of the kinematics for the action.
act_kin_start_unit	CHAR	The units for the start time.
act_kin_dur_val	INTEGER	The duration of the kinematics for the action.
act_kin_dur_unit	CHAR	The units for the duration.

f. act_manner

A description of the manner of the action based on Laban Movement Observation.

Column name	**Data type**	**Description**
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
mann_space	FLOAT	Indirect = -1, Direct = 1
mann_weight	FLOAT	Light = -1, Strong = 1
mann_time	FLOAT	Sustained = -1, Sudden = 1
mann_flow	FLOAT	Free = -1, Bound = 1
mann_vert	FLOAT	Vertical: Sinking = -1, Rising = 1
mann_lat	FLOAT	Lateral: Narrowing = -1, Widening = 1
mann_sagit	FLOAT	Sagittal: Retreating = -1, Advancing = 1
mann_shapefl	FLOAT	Shape Flow

g. action_status

This table holds the current status for an action. It is mainly used for after action reporting of what happened during the simulation.

Column name	**Data type**	**Description**
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
status_id	SMALLINT	An indicator of the current status of the action. This links to the statuses in the PAR table: `failed = 0, preempted = 1, suspended = 2, completed = 3, executing = 4.`

h. adverb_exp

These fields hold adverbs and adverb modifiers that might be used to describe the action. These might then be used by the corresponding motion generator to modify the action performance accordingly. They might also effect planning (were it to exist).

Column name	**Data type**	**Description**
act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. [PRIMARY KEY]
adverb_name	VARCHAR	An adverb that would need to have meaning associated with it in a planner or motion generator.

adverb_mod_name	VARCHAR	An adverb modifier that would need to have meaning associated with it in a planner or motion generator. For example, “move more quickly”

i. const_graph

This table is used to describe sub-actions and their execution.

Column name	Data type	Description
const_act_id	SMALLINT	The action id that is used to link this PAR to other tables including the main action table. This is the original complex action’s id. [PRIMARY KEY]
const_sub_act_id	SMALLINT	This is one of the sub-action id’s.
const_pcg_type	CHAR	This is the type of constraint in the PAR Constraint Graph. It could be Sequential (s), Parallel (p), Jointly Parallel (j), Independently Parallel (i), or While Parallel (w).
const_order	TINYINT	This is the order of the actions if needed for the type of constraint.

For example:

const_act_id	const_sub_act_id	const_pcg_type	const_order
0	10	w	1
0	11	w	2
0	12	p	3
0	13	p	3
0	14	i	4

Sub-action 11 will only be performed while sub-action 10 is being performed. Then sub-actions 12 and 13 will be performed together with sub-action 14 for as long as either (12 and 13) or 14 is being performed. (I think). Other intermediate complex actions may have to be defined to get the desired (nested) behavior.

j. direction_spec

This table stores direction specifications.

Column name	Data type	Description
dir_id	SMALLINT	The direction id that is used to link this direction to other tables including the main action table. [PRIMARY KEY]
direction	VARCHAR	The direction name such as: across, after, against, ahead-of, along, apart, around, away, away-from, back, back-and-forth, backward, behind, by, clockwise, counterclockwise, down, downward, forward, from, in, in-the-direction-of, into, inward, off, off-of, on, onto, onward, out, out-of, outward, over, sideways, through, to, together, toward, under, up, up-and-down, upward, with. Also, left and right.

obj_id	SMALLINT	The id of the object associated with the direction.
dir_value	FLOAT	The number of units of the movement, such as 4 spaces to the left.
dir_unit	CHAR	A representation of the units of movement such as meters.

k. location_spec

This tables stores location specifications.

Column name	Data type	Description
loc_id	SMALLINT	The location id that is used to link this direction to other tables including the path table. [PRIMARY KEY]
position_id	SMALLINT	An id to link to the position specification table.
seq_num	INTEGER	Used to determine the sequence of positions in the location specification.

l. path

This table is used to describe the path that might be needed for an action. The path might also be specified/created in a motion generator or planner. There is likely more than one entry per path id to form an entire path.

Column name	Data type	Description
path_id	SMALLINT	The path id that is used to link this path to other tables including the main action table. [PRIMARY KEY]
path_start_pos_x	FLOAT	The path starting x position.
path_start_pos_y	FLOAT	The path starting y position.
path_start_pos_z	FLOAT	The path starting z position.
path start_orient_x	FLOAT	The path starting x orientation.
path_start_orient_y	FLOAT	The path starting y orientation.
path_start_orient_z	FLOAT	The path starting z orientation.
path_end_pos_x	FLOAT	The path ending x position.
path_end_pos_y	FLOAT	The path ending y position.
path_end_pos_z	FLOAT	The path ending z position.
path_end_orient_x	FLOAT	The path ending x orientation.
path_end_orient_y	FLOAT	The path ending y orientation.
path_end_orient_z	FLOAT	The path ending z orientation.
path_distance	FLOAT	The distance of this segment of the path.
path_dir_id	SMALLINT	An id to link to a direction specification.
path_start_loc_id	SMALLINT	An id to link to a location specification indicating the starting location of the action.
path_end_loc_id	SMALLINT	An id to link to a location specification indicating the ending location of the action.

path_mod_id	SMALLINT	An id to link to modifier specifications.
path_dir_num	INTEGER	There may be multiple path directions for an action and this is used to order them.

m. path_modifiers

This table is used to specify modifiers of paths.

Column name	Data type	Description
mod_id	SMALLINT	The modifier id that is used to link this time to other tables including the path table. [PRIMARY KEY]
modifer	VARCAR	The modifier term such as: FOLLOWING, GUIDING, SHADOWING, SWARMING, CONGREGATING, DISPERSING, etc.

n. position_spec

This table is used to specify positions.

Column name	Data type	Description
pos_id	SMALLINT	The position id that is used to link this time to other tables including the path table. [PRIMARY KEY]
position	VARCHAR	The position term such as: ON, AT, IN,
obj_id	SMALLINT	A id indicating the object referenced in the position term.

o. time_spec

This table is used to specify various times in PAR.

Column name	Data type	Description
time_id	SMALLINT	The time id that is used to link this time to other tables including the main action table to start times and durations. [PRIMARY KEY]
time_type	CHAR	This might be an absolute simulation time (a) or duration (d).
time_val	FLOAT	The value for the time specification.

2. Object Hierarchy

The following tables are mainly related to PAR objects and are meant to include both general object descriptions and specific objects. These also hold the current properties of the world.

a. obj_act

This table links PAR action with the PAR object participants.

Column name	Data type	Description
obj_id	SMALLINT	This is the associated object's id. There may be more than one object per action, meaning more than one object participating in a single action. Certainly there may also be more than one action id per object, meaning that a single object participates in more than one action. [PRIMARY KEY]
act_id	SMALLINT	This is the id of the action as referenced in the main action table.
obj_num	TINYINT	If an action has more than one object participant, then the order of the objects has implicit meaning about the type of participation. This field provides us with a means of determining that order aside from simple order in the table.

b. object

This is the main object table and is linked to many other object oriented tables.

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to many of the other tables in the database. [PRIMARY KEY]
obj_name	VARCHAR	The name of the object. The convention is that names with underscores and numbers are instances and those with out are the general "parents" or classes. For example, "Chair" would contain general properties of a Chair and "Chair_0" would contain information about an actual chair in the environment.
obj_agent	CHAR	Indicated whether or not this object is also an agent (can execute actions). y = yes, n = no or t = true, f = false
obj_bv_type	CHAR	This is the bounding volume type or shape. c = cylinder, b = box, etc
obj_coord_pos_x	FLOAT	The coordinate position in x.
obj_coord_pos_y	FLOAT	The coordinate position in y.
obj_coord_pos_z	FLOAT	The coordinate position in z.
obj_pos_x	FLOAT	The current x position of this object.
obj_pos_y	FLOAT	The current y position of this object.
obj_pos_z	FLOAT	The current z position of this object.
obj_vel_x	FLOAT	The current velocity of this object in x.
obj_vel_y	FLOAT	The current velocity of this object in y.
obj_vel_z	FLOAT	The current velocity of this object in z.
obj_acc_x	FLOAT	The current acceleration of this object in x.

obj_acc_y	FLOAT	The current acceleration of this object in y.
obj_acc_z	FLOAT	The current acceleration of this object in z.
obj_orient_x	FLOAT	The current orientation of this object around x.
obj_orient_y	FLOAT	The current orientation of this object around y.
obj_orient_z	FLOAT	The current orientation of this object around z.
obj_parent_id	SMALLINT	The id of the parent object so that it can inherit the properties.
obj_part_of_id	SMALLINT	The id of the object that this object is a part of. For example a doorknob might be a part of a door.
obj_possessedBy_id	SMALLINT	The id of the object (or agent) that this object is possessed by. This can be used to provide default locations and objects for actions.

c. obj_boundvol

This table holds the bounding volume of an object

Column name	**Data type**	**Description**
obj_id	SMALLINT	The obj_id that is used to link the information in this table to other tables including the main object table. [PRIMARY KEY]
obj_bv1_x-obj_bv8_x	FLOAT	The x coordinates for points of the bounding volume.
obj_bv1_y-obj_bv8_y	FLOAT	The y coordinates for the points of the bounding volume.
obj_bv1_z-obj_bv8_z	FLOAT	The z coordinates for the points of the bounding volume.

d. obj_capable

This table associates objects with capabilities. For agents this is actions they are capable of performing (checked in applicability conditions). For other objects these are actions that object might be used in or participate in. Note that there may be many entries for each object and action.

Column name	**Data type**	**Description**
obj_id	SMALLINT	The obj_id that is used to link the information in this table to other tables including the main object table. [PRIMARY KEY]
action_id	SMALLINT	Id's for the actions that the objects can participate in.

e. obj_contents

This table lists the contents of an object (such as a room). There many be many entries for each object.

Column name	**Data type**	**Description**
obj_id	SMALLINT	The obj_id that is used to link the information in this table to other tables in the database including the main object table.

		[PRIMARY KEY]
obj_content_id	SMALLINT	The id of an object that is located in obj_id

f. obj_desc

The table where a description of the object can be found

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_descript	VARCHAR	An English description of the object. Could be used for explanations, etc.

g. obj_dir

The table can be used to describe various types of directions.

Column name	Data type	Description
dir_id	SMALLINT	An id for this direction. [PRIMARY KEY]
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database.
dir_type	CHAR	This is the type of direction r = relative, s = special
dir_name	VARCHAR	The name of the direction: front, back, left, right, along, inside, etc.
dir_pos_x	FLOAT	The x position of the direction.
dir_pos_y	FLOAT	The y position of the direction.
dir_pos_z	FLOAT	The z position of the direction.
dir_orient_x	FLOAT	The x orientation.
dir_orient_y	FLOAT	The y orientation.
dir_orient_z	FLOAT	The z orientation.

h. obj_location

The location of an object is another object that this object is in the contents of.

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_pid	SMALLINT	The id of the object that is the location of the object with id obj_id. [PRIMARY KEY]

i. obj_possessions

This table associates objects with their possessions. This would normally hold for agents. There may be many possessions for each object. This can help with default locations and objects for certain actions. resource allocation.

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_possess_id	SMALLINT	The object id for an object possessed by the object represented by the value in obj_id

j. obj_posture

This table associates objects with postures. These might be human postures (e.g. standing, prone, supine, etc) or object postures (e.g. upright, open, etc).

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
posture_name	VARCHAR	The name of the posture which would be associated with actual meaning in a planner or motion generator.

k. obj_prop

This table associates objects with properties. Properties can be rather general or generic. The meanings of the properties and their values come from how they are processed by functions, motion generators and planners.

Column name	Data type	Description
prop_id	SMALLINT	The prop_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
prop_name	VARCHAR	The name of the property.
prop_value	VARCHAR	The value of the property for this object.

l. obj_status

This table associates objects with their statuses. The meanings of the statuses would come from how they are processed by functions, motion generators and planners.

Column name	Data type	Description
status_id	SMALLINT	The status_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_id	SMALLINT	The obj_id that is used to link the information in this

		table to the other tables in the database. [PRIMARY KEY]
status_name	VARCHAR	The name of the status current status. For example, on or off.

m. obj_trash

This table holds the id of every object (and agent) that was dynamically created during the simulation. This allows us to clean up (remove them) database before the next simulation run.

Column name	Data type	Description
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
obj_id2	SMALLINT	NOT USED. It seems that I couldn't create a database with just one column.

n. site

This table associates objects with sites. In essence, sites are oriented points that can be used by motion generators to enable agents to interact with objects.

Column name	Data type	Description
site_id	SMALLINT	An id for this site [PRIMARY KEY]
obj_id	SMALLINT	The obj_id that is used to link the information in this table to the other tables in the database. [PRIMARY KEY]
site_name	VARCHAR	The name of this site.
site_pos_x	FLOAT	The x position of the site.
site_pos_y	FLOAT	The y position of the site.
site_pos_z	FLOAT	The z position of the site.
site_orient_x	FLOAT	The x orientation of the site.
site_orient_y	FLOAT	The y orientation of the site.
site_orient_z	FLOAT	The z orientation of the site.
site_type_id	SMALLINT	The type of site represented. This links to actions to determine the placement of an agent relative to the object for doing the action. For example, where to stand when cleaning vs about to sit for a chair.

3. Others (used for crowd simulations)

a. groups

A table holding information about agent groups. This table is linked to the members table. Group id's are also found in the action table to assign actions to groups of agents.

Column name	Data type	Description

group_id	SMALLINT	An id for this group. [PRIMARY KEY]
group_quantity	INT	The number of agents that should be in this group.
group_name	VARCHAR	The name of the group. If the name is of the form Role_#, then the group is assumed to be linked to a role and members of the group are assigned that role.

b. members

A table indicating which agents are in which groups.

Column name	Data type	Description
group_id	SMALLINT	The group id. [PRIMARY KEY]
agent_id	SMALLINT	The object id of the agent that is a member of the group indicated by group_id.

c. need_fullfill

This table links needs with actions and objects that can be used to fulfill the needs.

Column name	Data type	Description
need_id	SMALLINT	The id of the need which [PRIMARY KEY]
act_id	SMALLINT	An action that can be used to fulfill the need.
obj_id	SMALLINT	An object that can be a participant of the action to fulfill the need.
growth_rate	FLOAT	The amount the need is fulfilled by doing the action with the object. Needs correspond to units/hour normalized to 0 for empty and 1 for full.

d. need_states

This table indicates what the current need levels are for each of the agents. There will be multiple entries for each need as well as for each agent.

Column name	Data type	Description
need_id	SMALLINT	An id for this need [PRIMARY KEY]
agent_id	SMALLINT	The id of the agent
level	FLOAT	The current level of the need. Needs correspond to units/hour normalized to 0 for empty and 1 for full.

e. needs

This table associates need ids with need names and decay rates. Ultimately, this may change to indicate different decay rates for different agents.

Column name	Data type	Description

need_id	SMALLINT	An id for this need. [PRIMARY KEY]
need_name	VARCHAR	A human readable name for the need.
decay_rate	FLOAT	How fast the need decays. Needs correspond to units/hour normalized to 0 for empty and 1 for full.

f. reacting

This table holds what reactions agents have had along with the objects and times. It is used to keep a record of reactions and also to ensure that agents don't react to the same stimuli in quick succession.

Column name	**Data type**	**Description**
reaction_id	SMALLINT	An id for this reaction. [PRIMARY KEY]
agent_id	SMALLINT	The agent that reacted.
action_id	SMALLINT	The action the agent performed.
object_id	SMALLINT	The object (or agent) that was reacted to.
time_id	SMALLINT	Links to the time_spec table to indicate when the reaction took place.

g. reactions

This table is used to create reactions. It can represent many different scenarios. Individual reactions to an object. Individual reactions to a type of object. Individual reactions to a group of people. A group that reacts to an object. A group that reacts to a type of object. A group that reacts to a group. All agents reacting to various things.

Column name	**Data type**	**Description**
reaction_id	SMALLINT	An id for this reaction [PRIMARY KEY]
agent_id	SMALLINT	The id of the agent that should react or the id of the group if it is a group reaction as indicated in groupB.
groupB	SMALLINT	Indicates whether this is a group reaction or individual reaction. If groupB = 0, then agent_id corresponds to an object_id for a single agent. If groupB = 1, then agent_id corresponds to a group id as found in the group table. If groupB = 2, then all agents in the simulation are assigned this reaction.
action_id	SMALLINT	The action to be performed when the stimuli is observed.
object_id	SMALLINT	The stimuli for the reaction. This could be a specific object id or a parent or type id. It might also be the id of a group as found in the group table. If it is a group id, then react_to_group will be 1.
posture_id	SMALLINT	If they are to react to objects or agents with a certain posture, that can be indicated here. (NOT

		CURRENTLY IMPLEMENTED)
property_id	SMALLINT	If they are to react to objects or agents with a certain property, that can be indicated here. (NOT CURRENTLY IMPLEMENTED)
status_id	SMALLINT	If they are to react to objects or agents with a certain status, that can be indicated here. (NOT CURRENTLY IMPLEMENTED)
react_to_group	SMALLINT	Is this a reaction to a group of agents? 0 for no and 1 for yes. If yes, then object_id actually contains a group id.

h. role

This table defines roles for agents. It associates role names with id and can be used to indicate default actions for the role and when appropriate default locations

Column name	**Data type**	**Description**
role_id	SMALLINT	An id for this role. [PRIMARY KEY]
role_name	VARCHAR	A human readable name for the role.
default_act_id	SMALLINT	If agents with this role have nothing to do, what should they do by default.
obj_id	SMALLINT	This field indicates a location resource that an agent with this role should have. For example, a professor should have an office. The resource allocate will then try to allocate one for each professor.

i. role_assignments

This table can be used to assign specific agents to specific roles. Such assignments can be made for groups through the group table.

Column name	**Data type**	**Description**
role_id	SMALLINT	An id for this role [PRIMARY KEY]
obj_id	SMALLINT	The id of the agent being assigned the role.