Appendix A: DIRM attributes

	Attribute	Type	Description	Assignment	Changes per tick	Things to note
Survivor agents	survival-pts	int	Survival points are used to track resources that a survivor needs to sustain life.	Everyone starts at 100.	This will become the survivor's first priority to fulfill. However, right after a disaster, survival points are not an issue, because they emerge from the disaster at 100 percent. This becomes a priority when the agent has less than two days of survival supplies remaining. The value is calculated by using the previous value minus the agent's cost-per-day variable.	Death occurs when the agent has less than 5 points left. This is because when less than 5 points were used agents that were close to zero were not always dying.
	recovery-pts	int	Recovery points are used to track the recovery of an agent.	Everyone starts at 100, but during the set-up process a disaster occurs. The disaster takes a bit of the recovery points from all survivors. The amount taken is determined by the distribution of damage, the mean of the damage, and in the case of a normal distribution -- the standard deviation of damage is also taken into account.	After the intial disaster, no survivor ever should lose recovery points. If they have their survival needs met, then their goal is to search for resources to help them recover.	
	recovered?	bool	Has the survivor recovered?	Upon set-up, this is set to false.	With each tick that passes, the recovery-points are checked. If the recovery-points are at 100, then this changes to True.	When the agent is recovered, it then heads back home to enjoy its new life.
	survivor-days	int	This is the number of days that a survivor can survive without any additional resources.	This value is established by agent upon set-up. Agents are assigned the number of days that they would survive wtihout resources based on the average value of three days with a standard deviation of one day to account for those who are in better or worse health.	Value is constant.	
	cost-per-day	int	This is the cost of survival points per day.	This value is established at set up and is calculated in the following way: cost-per-day = (100 / survivor-days)	Value is constant.	
	cost-per-tick	int	This is the cost of survival points per tick.	This value is established at set up and is calculated in the following way: cost-per-tick = (cost-per-day / 16)	Value is constant.	
	distance-traveled	int	This is the distance that an agent travels throughout the model run.	Everyone starts at 0, because their initial starting point is their home patch.	Every time the survivor steps, this value increases by one. An agent will take a step towards some action if they are not in the middle of a transaction or if they are not 100 percent recovered and at home.	
	age	int	This is the age of the agent.	Everyone starts at 0.	With each tick that passes, everyone increases by one. If the turtle dies, then their value ceases to increase.	
Helper agent	h-type	int	H-type is the type of supplies the helper is carrying.	Helpers are randomly assigned an h-type of 0 or 1.	Value is constant.	It is assumed that the population of helpers is 50 percent carriers of survival supplies and 50 percent carriers of recovery supplies.
	mobility	bool	Mobility is the designation of whether an agent is mobile or not. If the agent is mobile, then they are able to move about the model. If they are not, then they have to start at their home patch.	Upon set-up, the system takes the %-helpers-mobile and assigns a mobility of True or False according to the percentage.	Value is constant.	If agent is mobile, to encourage them to not be located at their distribution point, the agents are scattered at random away from their distribution point. However, they must return to this point to refill their supplies.
All agents	home-patch	patch	This is the home base of the agent. For a helper, this is a supply center. For a survivor this is a random spot in the world.	Upon set-up, this value is the value that the agent starts in. If the agent is a helper and they are mobile, then they are scattered away from this point.	Value is constant.	Agents will return to this point. Survivors will return home to drop off supplies. Helpers return to this point to fill up on supplies.
	current-need	int	This is either a 0, 1, or 2, which is assigned to identify the current need of the agent.	No value assigned upon set-up.	When a tick passes, each agent decides their priority and seeks out that priority. Helpers go home to refill on supplies, execute an exchange, or head in the direction of an agent in need. Survivors go home to unload supplies, look for survival supplies, or look for recovery supplies.	Helpers exchange for a tick, while survivors do not. The reason for this is that helpers are more likely to be stopped in the road by individuals needing supplies rather than a survivor who is independent and looking to fulfill their needs. If a helper exchanges with one survivor, it is unable to exchange with another until the next tick.
	capacity	int	This is the amount of supplies that both the helper and the survivor can carry.	The initial values are set by the user through the model controllers.	Value is constant.	Survivors can carry less than helpers, because it is assumed that helpers are more likely to have resources and capacity to carry more.
Global	center-patches	patches	These are patches where centers are located.	Upon set-up, a certain number of patches in random locations become supply distribution centers.	Value is constant.	

Appendix A: DIRM attributes

movement-fwd	int	This is the value that the agent moves forward with each step. Currently, this set to one, meaning an agent can move forward one patch per turn as opposed to .5 or two.	Default is one, no action is taken upon set-up.	Value is constant.
capacity-setting	int	This is the value of the final helper capacity. If there are enough supplies in the system, then it is set to the helper-supply-capacity. If there are not enough supplies in the system, then it is set to - (total number of supplies / number of helpers).	This is calculated upon set-up and is assigned as a uniform value.	Value is constant.