bad_weather__conditions(t) = bad_weather__conditions(t - dt) + (increasing_precipitation - decreasing_precipitation) * dt
INIT bad_weather__conditions = 0
INFLOWS:
increasing_precipitation = precipitation/10
OUTFLOWS:
decreasing_precipitation = delay(increasing_precipitation,1,0)

evacuated_people(t) = evacuated_people(t - dt) + (conducting_measures_evacuation) * dt
INIT evacuated_people = 0
INFLOWS:
conducting_measures_evacuation = (smth1(number_of_relief_units_conducting_evacuation*performance_per_hour_and_person_evak,delay_due_ot__lacking_mobility+delay_due_to_lacking_cooperation_population+delay_due_to_transportation+delay_due_to_lacking_cooperation_relief_2,0))

mobile_protection_installed(t) = mobile_protection_installed(t - dt) + (conducting_measures_mobile_protection) * dt
INIT mobile_protection_installed = 0
INFLOWS:
conducting_measures_mobile_protection = (SMTH1(number_of_relief_units_mobile_protection_measures*performance_per_hour__and_person_mobile_protection,delay_due_to_transportation+delay_due_to_lacking_cooperation_relief,0))

mobile_protection_needed(t) = mobile_protection_needed(t - dt) + (-conducting_measures_mobile_protection) * dt
INIT mobile_protection_needed = inital_need_mobile_protection
OUTFLOWS:
conducting_measures_mobile_protection = (SMTH1(number_of_relief_units_mobile_protection_measures*performance_per_hour__and_person_mobile_protection,delay_due_to_transportation+delay_due_to_lacking_cooperation_relief,0))

number_of_people_that_have_to_be_evacuated(t) = number_of_people_that_have_to_be_evacuated(t - dt) + (-conducting_measures_evacuation) * dt
INIT number_of_people_that_have_to_be_evacuated = initial_need_evacuation

OUTFLOWS:
conducting_measures_evacuation = (smth1(number_of_relief_units_conducting_evacuation*performance_per_hour_and_person_evak,delay_due_ot__lacking_mobility+delay_due_to_lacking_cooperation_population+delay_due_to_transportation+delay_due_to_lacking_cooperation_relief_2,0))

number_of_relief_units_conducting_evacuation(t) = number_of_relief_units_conducting_evacuation(t - dt) + (relief_units_allocated_to_evacuation - back_to_stock_from_evacuation) * dt
INIT number_of_relief_units_conducting_evacuation = 0
INFLOWS:
relief_units_allocated_to_evacuation = (SMTH1(discrepancy/performance_per_hour_and_person_evak,(time_needed_for__decision_evacuation+delay_due_to_transportation),0))
OUTFLOWS:
back_to_stock_from_evacuation = number_of_relief_units_conducting_evacuation/12

number_of_relief_units_mobile_protection_measures(t) = number_of_relief_units_mobile_protection_measures(t - dt) + (relief_units_allocated_to_mobile_protection - back_to_stock_from_mobile_protection) * dt
INIT number_of_relief_units_mobile_protection_measures = 0
INFLOWS:
relief_units_allocated_to_mobile_protection = ((SMTH1(gap/performance_per_hour__and_person_mobile_protection,(time_needed_for_decision_mobile_protection+delay_due_to_transportation),0)))
OUTFLOWS:
back_to_stock_from_mobile_protection = number_of_relief_units_mobile_protection_measures/12

relief_units_stock(t) = relief_units_stock(t - dt) + (back_to_stock_from_mobile_protection + back_to_stock_from_evacuation - relief_units_allocated_to_mobile_protection - relief_units_allocated_to_evacuation) * dt
INIT relief_units_stock = initial_relief_stock
INFLOWS:
back_to_stock_from_mobile_protection = number_of_relief_units_mobile_protection_measures/12
back_to_stock_from_evacuation = number_of_relief_units_conducting_evacuation/12
OUTFLOWS:
relief_units_allocated_to_mobile_protection = ((SMTH1(gap/performance_per_hour__and_person_mobile_protection,(time_needed_for_decision_mobile_protection+delay_due_to_transportation),0)))
relief_units_allocated_to_evacuation = (SMTH1(discrepancy/performance_per_hour_and_person_evak,(time_needed_for__decision_evacuation+delay_due_to_transportation),0))

water_level_stock(t) = water_level_stock(t - dt) + (inflow_water_level - outflow_water_level) * dt
INIT water_level_stock = 0
INFLOWS:
inflow_water_level = water_level*(0.5+increase_of_water_level)
OUTFLOWS:
outflow_water_level = delay(inflow_water_level,1,0)

cooperation_witihin_the_relief_units = GRAPH(number_of_relief_units_mobile_protection_measures)
(0.00, 0.00), (50.0, 0.105), (100, 0.225), (150, 0.32), (200, 0.425), (250, 0.555), (300, 0.67), (350, 0.74), (400, 0.81), (450, 0.88), (500, 1.00)

cooperation_witihin_the_relief_units_2 = GRAPH(number_of_relief_units_conducting_evacuation)
(0.00, 0.00), (50.0, 0.18), (100, 0.5), (150, 0.98), (200, 1.30), (250, 1.60), (300, 1.92), (350, 2.28), (400, 2.70), (450, 3.26), (500, 3.94)

delay_due_ot__lacking_mobility = GRAPH(mobility_of_the__population)
(0.00, 0.535), (0.1, 0.903), (0.2, 1.22), (0.3, 1.52), (0.4, 1.80), (0.5, 2.18), (0.6, 2.36), (0.7, 2.83), (0.8, 3.28), (0.9, 3.81), (1, 3.98)

delay_due_to_flooded_road = GRAPH(flooded_road)
(0.00, 0.64), (0.1, 0.675), (0.2, 0.762), (0.3, 0.99), (0.4, 1.29), (0.5, 1.90), (0.6, 2.95), (0.7, 3.46), (0.8, 3.74), (0.9, 3.93), (1, 3.95)

delay_due_to_lacking_cooperation_population = GRAPH(not_cooperative_share_of_population)
(5.00, 0.535), (5.50, 0.798), (6.00, 1.02), (6.50, 1.20), (7.00, 1.64), (7.50, 2.02), (8.00, 2.39), (8.50, 2.88), (9.00, 3.39), (9.50, 3.76), (10.0, 4.00)

delay_due_to_lacking_cooperation_relief = GRAPH(cooperation_witihin_the_relief_units)
(0.00, 0.02), (0.1, 0.32), (0.2, 0.62), (0.3, 1.00), (0.4, 1.32), (0.5, 1.68), (0.6, 1.98), (0.7, 2.34), (0.8, 2.78), (0.9, 3.24), (1, 4.00)

delay_due_to_lacking_cooperation_relief_2 = GRAPH(cooperation_witihin_the_relief_units_2)
(0.00, 0.00), (0.1, 0.125), (0.2, 0.215), (0.3, 0.295), (0.4, 0.365), (0.5, 0.47), (0.6, 0.55), (0.7, 0.64), (0.8, 0.725), (0.9, 0.825), (1, 1.00)

delay_due_to_road_condition = GRAPH(road_condition)
(0.00, 4.00), (0.1, 4.00), (0.2, 3.93), (0.3, 3.58), (0.4, 3.06), (0.5, 2.00), (0.6, 1.36), (0.7, 0.99), (0.8, 0.798), (0.9, 0.64), (1, 0.5)

delay_due_to_transportation = (delay_due_to_flooded_road*Threshold_1+delay_due_to_flooded_road*Threshold_2)+(delay_due_to_road_condition)

discrepancy = if water_level_stock >700 then (number_of_people_that_have_to_be_evacuated-evacuated_people) else 0
flooded_road = GRAPH(water_level_stock)
(700, 0.52), (750, 0.53), (800, 0.555), (850, 0.585), (900, 0.618), (950, 0.657), (1000, 0.708), (1050, 0.768), (1100, 0.87), (1150, 0.955), (1200, 1.00)

gap = if water_level_stock > 500 then (mobile_protection_needed-mobile_protection_installed) else 0
increase_of_water_level = 0.5
inital_need_mobile_protection = 1000
initial_need_evacuation = 1000
initial_relief_stock = 50
mobility_of_the__population = random(0,1)
not_cooperative_share_of_population = random(5,10)
performance_per_hour_and_person_evak = LOGNORMAL(10,10*0.2)
performance_per_hour__and_person_mobile_protection = LOGNORMAL(4.5,4.5*0.2)
precipitation = GRAPH(TIME)
(0.00, 0.00), (1.01, 0.00), (2.02, 0.00), (3.03, 0.00), (4.04, 0.00), (5.05, 0.00), (6.06, 0.00), (7.07, 0.00), (8.08, 0.00), (9.09, 0.00), (10.1, 0.00), (11.1, 0.00), (12.1, 0.00), (13.1, 0.00), (14.1, 0.00), (15.2, 0.00), (16.2, 0.00), (17.2, 0.00), (18.2, 0.00), (19.2, 0.00), (20.2, 10.0), (21.2, 6.00), (22.2, 1.00), (23.2, 0.00), (24.3, 3.00), (25.3, 10.0), (26.3, 5.00), (27.3, 0.00), (28.3, 0.00), (29.3, 27.0), (30.3, 7.00), (31.3, 1.00), (32.3, 0.00), (33.3, 0.00), (34.4, 6.00), (35.4, 38.0), (36.4, 8.00), (37.4, 4.00), (38.4, 0.00), (39.4, 4.00), (40.4, 9.00), (41.4, 3.00), (42.4, 26.0), (43.5, 9.00), (44.5, 2.00), (45.5, 6.00), (46.5, 4.00), (47.5, 0.00), (48.5, 0.00), (49.5, 0.00), (50.5, 0.00), (51.5, 0.00), (52.5, 0.00)...
road_condition = GRAPH(bad_weather__conditions)
(0.00, 1.00), (0.75, 0.915), (1.50, 0.835), (2.25, 0.77), (3.00, 0.695), (3.75, 0.59), (4.50, 0.47), (5.25, 0.34), (6.00, 0.23), (6.75, 0.13), (7.50, 0.00)

Threshold_1 = IF (water_level_stock < 700) then 0 else 1
Threshold_2 = IF (water_level_stock >1200) THEN 1 ELSE 0
time_needed_for_decision_mobile_protection = random(0.5,2)
time_needed_for__decision_evacuation = random(0.5,2)
water_level = GRAPH(TIME)
(0.00, 400), (1.01, 409), (2.02, 424), (3.03, 445), (4.04, 475), (5.05, 501), (6.06, 532), (7.07, 549), (8.08, 572), (9.09, 589), (10.1, 609), (11.1, 619), (12.1, 634), (13.1, 644), (14.1, 648), (15.2, 650), (16.2, 651), (17.2, 652), (18.2, 653), (19.2, 655), (20.2, 655), (21.2, 656), (22.2, 656), (23.2, 658), (24.3, 660), (25.3, 662), (26.3, 663), (27.3, 663), (28.3, 663), (29.3, 661), (30.3, 660), (31.3, 659), (32.3, 659), (33.3, 660), (34.4, 660), (35.4, 663), (36.4, 666), (37.4, 668), (38.4, 669), (39.4, 666), (40.4, 661), (41.4, 656), (42.4, 651), (43.5, 644), (44.5, 636), (45.5, 632), (46.5, 631), (47.5, 629), (48.5, 633), (49.5, 641), (50.5, 644), (51.5, 650), (52.5, 661)...