

Appendix H: Agent-Based Model Pseudo Code

This appendix provides essential pseudocode for the agent-based model implementation. Keywords are in UPPERCASE, indentation denotes scope, and // begins comments.

Global Parameters and Initialization

```
```\n\n// Simulation constants\nSIM_DURATION_YEARS ← 3           // Jan 2011 - Jan 2014\nDAYS_PER_MONTH ← 30\nDAYS_PER_PREGNANCY ← 270\nNUM_FACILITIES ← 3              // Facilities A, B, C\n\n// Key probabilities (calibrated)\nP_PREGNANT_MONTHLY ← 0.025       // Monthly conception rate\nP_LOST_PREGNANCY ← 0.20          // Miscarriage/stillbirth\nP_DELIVERY_COMPLICATION ← 0.46  // Delivery complication\n\n// Behavioral modifiers (calibrated)\nM_FACILITY_EXP ← 0.30            // Facility experience weight\nM_HOME_EXP ← 0.00                // Home experience weight\nM_SOCIAL ← 0.35                  // Social influence weight\nM_TBA ← 0.35                     // TBA incentive weight\n\nPROCEDURE initializeSimulation(seed)\n    RNG.setSeed(seed)
```

tDay  $\leftarrow$  0, tMonth  $\leftarrow$  1

// Create 3 facilities with baseline attributes from data

FOR facilityID IN {"A", "B", "C"} DO

    facility  $\leftarrow$  loadFacilityData(facilityID)

    \\calibrated capacity based on nominal beds

    facility.capacity  $\leftarrow$  facility.bedsNominal + 3

    ADD facility TO facilities

END FOR

// Generate women population (1,866 total)

FOR facility IN facilities DO

    numWomen  $\leftarrow$  calculateCatchmentSize(facility)

    FOR i  $\leftarrow$  1 TO numWomen DO

        woman  $\leftarrow$  new Woman

        woman.assignedFacility  $\leftarrow$  facility

        sampleDemographics(woman , facility.id) // Table 3 distributions

        IF RNG.uniform() < 0.15 THEN woman.state  $\leftarrow$  PREGNANT

        ELSE woman.state  $\leftarrow$  NON\_PREGNANT

        buildSocialNetwork(woman , 5) // 5 peers per woman (calibrated)

        ADD woman TO women

    END FOR

END FOR

manager  $\leftarrow$  new DistrictManager(facilities)

END PROCEDURE

```\n

Main Simulation Loop

```\n

```
PROCEDURE runSimulation()
 WHILE tDay < SIM_DURATION_YEARS * 365 DO
 // Daily agent updates
 FOR woman IN women PARALLEL DO
 advanceWomanState(woman)
 END FOR

 // Monthly events
 IF (tDay + 1) MOD DAYS_PER_MONTH == 0 THEN
 processMonthlyEvents()
 END IF

 tDay ← tDay + 1
 END WHILE
END PROCEDURE
```

```
PROCEDURE advanceWomanState(woman)
 SWITCH woman.state
 CASE NON_PREGNANT:
 IF RNG.bernoulli(P_PREGNANT_MONTHLY / DAYS_PER_MONTH) THEN
 woman.state ← PREGNANT
 woman.daysInState ← 0
```

END IF

CASE PREGNANT:

IF woman.daysInState >= DAYS\_PER\_PREGNANCY THEN

    deliveryProbability ← computeDeliveryChoice(woman)

    IF RNG.uniform() < deliveryProbability THEN

        processFacilityDelivery(woman)

    ELSE

        processHomeDelivery(woman)

    END IF

ELSE IF RNG.bernoulli(P\_LOST\_PREGNANCY/DAYS\_PER\_PREGNANCY) THEN

    woman.state ← NON\_PREGNANT // Pregnancy loss

END IF

CASE POSTPARTUM , COMPLICATION\_\*:

IF woman.daysInState >= 42 THEN

    woman.state ← NON\_PREGNANT

END IF

END SWITCH

    woman.daysInState ← woman.daysInState + 1

END PROCEDURE

...

[Delivery Decision Model](#)

...

FUNCTION computeDeliveryChoice(woman) RETURNS float

```
facility ← woman.assignedFacility
```

```
// Logistic regression (Binyaruka et al. model)
```

```
X ← [woman.education , woman.previousPregs , woman.ancVisits4Plus ,
 woman.travelTime , facility.drugAvailability , facility.kindness ,
 facility.interpersonalQuality , facility.outOfPocketCosts ,
 computeOccupancyFactor(facility)]
```

```
linearPredictor ← DOT_PRODUCT(X, LOGIT_COEFFICIENTS)
```

```
baseProbability ← EXP(linearPredictor) / (1 + EXP(linearPredictor))
```

```
// Apply behavioral modifiers
```

```
socialInfluence ← computeSocialInfluence(woman.socialPeers)
```

```
modifierSum ← woman.facilityExp * M_FACILITY_EXP +
 woman.homeExp * M_HOME_EXP +
 socialInfluence * M_SOCIAL +
 facility.tbaIncentives * M_TBA
```

```
// Scaling transformation
```

```
IF modifierSum < 0 THEN
```

```
 finalProbability ← baseProbability * (1 + modifierSum)
```

```
ELSE
```

```
 finalProbability ← baseProbability *
 (1 + modifierSum * (1 - baseProbability))
```

```
END IF
```

```
RETURN CLAMP(finalProbability , 0, 1)

END FUNCTION

PROCEDURE processFacilityDelivery(woman)

 facility ← woman.assignedFacility

 // Check capacity
 IF facility.bedsOccupied >= facility.capacity THEN
 processOutOfSystemDelivery(woman)
 RETURN
 END IF

 // Process delivery
 facility.bedsOccupied ← facility.bedsOccupied + 1
 facility.birthsInFacilityMonth ← facility.birthsInFacilityMonth + 1
 facility.birthsCatchmentMonth ← facility.birthsCatchmentMonth + 1

 // Update woman experience based on service quality
 experienceScore ← 0.4 * facility.drugAvailability +
 0.4 * facility.kindness + 0.2 * facility.gifts
 woman.facilityExp ← IF RNG.uniform() < experienceScore THEN +1
 ELSE -1

 // Handle complications
 IF RNG.bernoulli(P_DELIVERY_COMPLICATION) THEN
 woman.state ← COMPLICATION_FACILITY
```

```
 facility.complicationReferrals ←
 facility.complicationReferrals + 1
ELSE
 woman.state ← POSTPARTUM
END IF
 woman.daysInState ← 0
END PROCEDURE
` ` `
```

#### Monthly Events and P4P Implementation

```
` ` `
PROCEDURE processMonthlyEvents()
 // Update facility metrics
 FOR facility IN facilities DO
 updateFacilityOccupancy(facility)
 resetDailyCounters(facility)
 END FOR

 // P4P performance evaluation (every 6 months)
 IF tMonth IN {6, 12, 18, 24, 30, 36} THEN
 evaluatePerformance()
 END IF

 // Bonus payments (3 months after evaluation)
 IF tMonth IN {9, 15, 21, 27, 33, 39} THEN
 processBonusPayments()
```

END IF

// District manager visits (every 3 months)

IF tMonth MOD 3 == 0 THEN

    processManagerVisits()

END IF

tMonth  $\leftarrow$  tMonth + 1

END PROCEDURE

PROCEDURE evaluatePerformance()

FOR facility IN facilities DO

    // Calculate 6-month facility-based delivery rate

    currentFB  $\leftarrow$  SUM\_LAST\_6\_MONTHS(facility.birthsInFacility) /  
        SUM\_LAST\_6\_MONTHS(facility.birthsCatchment)

    previousFB  $\leftarrow$  facility.lastPeriodFB

    indicator  $\leftarrow$  currentFB - previousFB

    // Determine bonus target (Equation 2)

    IF previousFB < 0.2 THEN target  $\leftarrow$  0.15

    ELSE IF previousFB < 0.4 THEN target  $\leftarrow$  0.10

    ELSE IF previousFB < 0.85 THEN target  $\leftarrow$  0.05

    ELSE target  $\leftarrow$  0.0 // Maintain performance

    // Award bonus

    IF indicator  $\geq$  target THEN



```
 facility.bonusEarned ← 1.0
ELSE IF indicator >= 0.75 * target THEN
 facility.bonusEarned ← 0.5
ELSE
 facility.bonusEarned ← 0.0
END IF

 facility.lastPeriodFB ← currentFB
END FOR
END PROCEDURE

PROCEDURE processBonusPayments()
 FOR facility IN facilities DO
 IF facility.bonusEarned > 0 AND NOT facility.bonusDelayed THEN
 applyBonusEffects(facility , facility.bonusEarned)
 ELSE IF facility.bonusDelayed THEN
 applyDelayEffects(facility) // Deterioration during delays
 END IF
 END FOR
END PROCEDURE

PROCEDURE applyBonusEffects(facility , bonusAmount)
 SWITCH bonusAmount
 CASE 1.0: // Full bonus
 facility.drugAvailability ← 1.0
 facility.kindness ← MAX(facility.kindness , 0.9)
```

```
facility.interpersonalQuality ← 1.0
facility.outOfPocketCosts ← facility.outOfPocketCosts * 0.8
facility.tbalIncentives ← 1
facility.outreachActive ← 1
facility.gifts ← 1
```

CASE 0.5: // Half bonus

```
facility.drugAvailability ←
 MAX(facility.drugAvailability , 0.8)
facility.kindness ← MAX(facility.kindness , 0.8)
facility.interpersonalQuality ←
 MAX(facility.interpersonalQuality , 0.8)
facility.outOfPocketCosts ← facility.outOfPocketCosts * 0.9
facility.tbalIncentives ← 1
facility.outreachActive ← RNG.bernoulli(0.5)
```

END SWITCH

END PROCEDURE

...

### District Manager Activities

...

PROCEDURE processManagerVisits()

FOR facility IN facilities DO

// Motivational boost (50% chance)

IF RNG.bernoulli (0.5) THEN

facility.kindness ← MIN(facility.kindness + 0.10, 1.0)

```
 facility.interpersonalQuality ← MIN(facility.IPQ + 0.10, 1.0)
 scheduleReversion(facility , tDay + 45) // Temporary effect
END IF

// Mitigate payment delays
IF facility.bonusDelayed THEN
 restorePreDelayLevels(facility)
END IF

// Drug sharing between facilities
IF facility.drugAvailability < 0.5 THEN
 donor ← findDonorFacility(facilities , 0.6) // Above 60%
 IF donor != NULL THEN
 transferAmount ← 0.10
 donor.drugAvailability ←
 donor.drugAvailability - transferAmount
 facility.drugAvailability ←
 facility.drugAvailability + transferAmount
 END IF
END IF

// Share best practices (if facility earned bonus)
IF facility.bonusEarned > 0 THEN
 adoptBestPractices(facility , facilities)
END IF
END FOR
```

END PROCEDURE

...

### Model Execution and Output

...

// Main execution loop for multiple replications

FOR replication ← 1 TO 500 DO

    seed ← MASTER\_SEED + replication

    initializeSimulation(seed)

    runSimulation()

    collectOutputs(replication)

    cleanupMemory()

END FOR

// Key outputs collected monthly for each facility:

// - Facility-based delivery rate

// - Drug availability , kindness , interpersonal quality

// - Out-of-pocket costs , occupancy rate

// - Bonus payments and delays

// - Manager interventions (drug sharing , strategy sharing)

// Scenario analysis

scenarios ← ["No\_P4P", "P4P\_with\_delays", "P4P\_without\_delays",  
              "P4P\_no\_manager\_visits"]

FOR scenario IN scenarios DO

    configureScenario(scenario)

```
runMultipleReplications(500)
calculateScenarioStatistics()
END FOR
` ` `
```