Appendix: Modelling Contextualized Reasoning in Complex Societies with “Endorsements”

Shah Jamal Alam, Armando Geller, Ruth Meyer and Bogdan Werth

In this appendix, Table 1 presents the pseudocode for the model’s main simulation loop, followed by a list of the model parameters and their default values given in Table A-2. Table A-3 provides the values for the three different types of exchanges. Table A-4 gives the pseudocode for the key modules and processes in the model. Figures S-1, S-2, and S-3 give further simulation results presented in Section 5.3 of the main article. Figure S-4 shows a reduced UML class diagram of key classes in the current version of the project source code. Figure S-5 shows the activity diagram for a person’s update functions respectively.

In the initialization phase, we first populate the simulated village by creating the joint- and nuclear-type households. For each household, a head and the rest of household members are created and assigned. Households are then placed on a 2D grid representing the physical space in the model. Neighbourhood links are created for each household, based on the existence of other households within the given radius (a model parameter). This is followed by creating the initial kinship network, via the Watts-Strogatz network (Watts and Strogatz 1998) with a rewiring probability of 0.2 (an assumption). We then assign godparents and compadre ties between the children and the adult agents in the model. Next, we assign random shared activities (abstract activities) to the agents. Finally, each adult agent then updates the endorsements for its acquaintances based on the relationships mentioned above. It then identifies a subset of acquaintances as ‘friends’, based on its friendship-endorsement scheme.

In the main schedule described in Table A-1, all agents’ update functions are called (see Table A-4), followed by the removal of all agents who die at the current time step (week) during the simulation. Acquainted agents of the deceased are thus notified. New child agents are born to married female agents who are still within fertile age range. The number of births in the model is created based on the annual birth rate per 1000 persons for the region. For every single and eligible male agent, we find a female partner for “elopement” based on the elopement-rate (see Table A-4). For each eloped couple, time for getting married is then determined. If the current week coincides with the week set for marriage, then determine marriage godparents for the married couple. Households of the bride and groom endorse each other as “relatives”; wife moves to the husband’s parents’ house and acquaintances are invited for services for the wedding of the couple. Following the households’ update functions (Table A-4), households without members and accommodate dependents of dissolved household if no adult guardian is alive are removed from the simulation. For each adult agent, update dynamic endorsements including “friend” endorsements for the acquaintances based on the update-elopement-rate (a model parameter). For each adult agent, respond to the invitations concerning the rite-de-passage of child of an acquaintance, i.e. birth, marriage and death. Finally, for each adult agent, perform ordinary and occasionally fiesta/ritual services and goods exchange for this week.
Table A-1: Outline of the main flow of a simulation run in the Tzintzuntzan model.

**Initialization Phase:**
I. populate-village
II. create-neighbourhood-links
III. create-extended-family-links
IV. create-compadre-links
V. create-shared-activities
VI. For each agent $\in$ Adult-Agents
   a. update endorsements
   b. find friends

**Main Schedule: Runs for $n$ time steps.**
1. FOR each agent $\in$ Agents
   call agent.update(current-time)
2. FOR each agent $\in$ Dead-Agents
   call purge-agent(agent)
   FOR each agent $\in$ Acquaintances
   call send-invitation (agent)
3. FOR each married-female-agent $\in$ Potential-Mothers
   IF new-birth(married-female-agent)
   THEN create-New-Agent (married-female-agent)
   call find-marriage-godparents
   call purge-agent(agent)
   FOR each agent $\in$ Acquaintances
   send-invitation (agent)
4. call create-New-Agents
5. call elopements (current-time)
6. FOR each eloped-couplelet $\in$ Eloped-Couples
   IF (couple.time-for-marriage equals current-time)
   THEN call marry(couple)
   call find-marriage-god-parents (couple)
   FOR each agent $\in$ Invited-Agents
   call send-invitation (agent)
7. FOR each household $\in$ Households
   call household.update(current-time)
8. FOR each household $\in$ Households-without-Guardians-Members
   call purge-Household(household)
9. FOR each agent $\in$ Adult-Agents
   IF agent.should-update-Endorsements(current-time)
   THEN call agent.update-Endorsements
10. FOR each agent $\in$ Adult-Agents
    call do-Rite-de-Passage-Services(current-time)
11. FOR each agent $\in$ Adult-Agents
    IF agent.should-send-Routine-Exchange(current-time)
    THEN call agent.send-Routine-Exchange

END
Table A-2: The Tzintzuntzan model parameters and their default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult-age</td>
<td>Age determining adulthood</td>
<td>16 years</td>
</tr>
<tr>
<td>Female-marriage-age-limit</td>
<td>Marriage age limit for females</td>
<td>43 years</td>
</tr>
<tr>
<td>Male-marriage-age-limit</td>
<td>Marriage age limit for males</td>
<td>53 years</td>
</tr>
<tr>
<td>Fertility-age-limit</td>
<td>Fertility age limit for female agents</td>
<td>43 years</td>
</tr>
<tr>
<td>Birth-rate</td>
<td>Annual birth rate</td>
<td>47.3</td>
</tr>
<tr>
<td>Elopement-probability</td>
<td>Per week probability for an unmarried male agent to elope an unmarried female agent</td>
<td>0.005</td>
</tr>
<tr>
<td>Build-nuclear-house</td>
<td>Per week probability for a married couple to build their own nuclear house</td>
<td>0.0025</td>
</tr>
<tr>
<td>Recall-period-continuous-endorsement</td>
<td>Maximum recall period for evaluating continuous endorsements sampled from logistic distribution</td>
<td>10 years (average)</td>
</tr>
<tr>
<td>Endorse-reliable-recall</td>
<td>Max recall period for evaluating an agent as “reliable” based on the exchange history</td>
<td>1 year (average)</td>
</tr>
<tr>
<td>Endorse-same-age</td>
<td>Range for determining “same age” defined by the proportion of one’s age</td>
<td>+/- 25% of age</td>
</tr>
<tr>
<td>Friends-limit</td>
<td>Maximum friends for an agent at a time</td>
<td>Uniform(7,11)</td>
</tr>
<tr>
<td>Remove-random-friend</td>
<td>Remove a random friend from friends’ list</td>
<td>0.005</td>
</tr>
<tr>
<td>Mean-ordinary-exchange</td>
<td>Average time for an adult agent to send ordinary services/goods – sampled from Exponential distribution</td>
<td>Average 2 weeks</td>
</tr>
<tr>
<td>Mean-ritual-exchange</td>
<td>Average time for an adult agent to send ritualistic services/goods – sampled from Exponential distribution</td>
<td>Average 8 weeks</td>
</tr>
<tr>
<td>Proportion-ordinary-exchange</td>
<td>Fraction of neighbours and friends selected for ordinary services/goods by an adult agent – sorted by their endorsement values</td>
<td>Uniform(0.2,0.3)</td>
</tr>
<tr>
<td>Proportion-ritual-exchange</td>
<td>Fraction of compadre and relatives selected for ritual exchange by an adult agent – sorted by their endorsement values</td>
<td>Uniform(0.15,0.25)</td>
</tr>
<tr>
<td>Endorsement-update-rate</td>
<td>Per week probability for an adult agent to update its endorsements</td>
<td>0.3333</td>
</tr>
<tr>
<td>Base-range</td>
<td></td>
<td>Uniform(1,3)</td>
</tr>
<tr>
<td>Neighbourhood-density</td>
<td>Density factor for neighbourhood evaluation over the grid</td>
<td>2</td>
</tr>
<tr>
<td>Neighbourhood-radius</td>
<td>Radius for calculating neighbourhood over the grid arbitrary assumption</td>
<td>4</td>
</tr>
<tr>
<td>Grid-size</td>
<td></td>
<td>20×20</td>
</tr>
<tr>
<td>Initial-household-dist</td>
<td>Nuclear household with husband, wife and children</td>
<td>Normal(5,3)</td>
</tr>
<tr>
<td>Init-nuclear-households</td>
<td>Initial proportion of nuclear households</td>
<td>60%</td>
</tr>
<tr>
<td>Tag-evolution-probability</td>
<td></td>
<td>0.005</td>
</tr>
<tr>
<td>Tag-length</td>
<td>Length of integer string representing tag</td>
<td>11</td>
</tr>
<tr>
<td>Tag-base</td>
<td>Exponent base to calculate similarity</td>
<td>5</td>
</tr>
<tr>
<td>Husband-age-Dist</td>
<td>Husband’s age distribution for a married couple at the start.</td>
<td>Normal(35, 7)</td>
</tr>
<tr>
<td>Household-head-Dist</td>
<td>Age distribution of the household head at the start</td>
<td>Normal(55, 5)</td>
</tr>
<tr>
<td>Age-difference-couples</td>
<td>Age difference between a husband and wife at the start</td>
<td>Normal (8, 2)</td>
</tr>
<tr>
<td>New-house-build-time</td>
<td>Time (in weeks) taken by a married son to build a separate house for his nuclear family</td>
<td>Normal(72,24) – within [24, 120]</td>
</tr>
</tbody>
</table>
Table A-3: Exchange types and values in the Tzintzuntzan model.

<table>
<thead>
<tr>
<th>Exchange type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Good</td>
<td>Lognormal Distribution (0.3, 0.3)</td>
</tr>
<tr>
<td>Ordinary Service</td>
<td>Lognormal Distribution (0.3, 0.7)</td>
</tr>
<tr>
<td>Ritual Good</td>
<td>Lognormal Distribution (0.5, 0.7)</td>
</tr>
<tr>
<td>Ritual Service</td>
<td>Lognormal Distribution (0.5, 0.9)</td>
</tr>
<tr>
<td>Birth Service</td>
<td>Lognormal Distribution (0.5, 0.9)</td>
</tr>
<tr>
<td>Marriage Service</td>
<td>Lognormal Distribution (0.5, 0.9)</td>
</tr>
<tr>
<td>Godparent’s birth service</td>
<td>Lognormal Distribution (0.3, 0.7)</td>
</tr>
<tr>
<td>Godparent’s elopement/marriage service</td>
<td>Lognormal Distribution (0.4, 0.9)</td>
</tr>
<tr>
<td>Godparent’s death service</td>
<td>Lognormal Distribution (0.4, 0.9)</td>
</tr>
</tbody>
</table>

In Table A-4, we outline the pseudocode for the key processes in the model. A further detailed description of the model will be presented separately where societal transitions in Tzintzuntzan are being studied.

Table A-4: Pseudocode of key processes in the Tzintzuntzan model.

**Pseudocode: agent: update(current-time)**
1. IF (birth-month AND birth-week)
   THEN increment-age
2. IF (gender is female AND is adult)
   IF (marital-status is married AND age <= fertility-age
      AND time-since-last-delivery >= 1 year)
      THEN potential-mothers.add(agent)
   ELSE IF (marital-status is single AND age within marriage-age)
      THEN eligible-females.add(agent)
   ELSE IF (gender is male AND adult AND single AND age within marriage-age)
      THEN eligible-males.add(agent)
   ELSE IF (gender is male
      THEN IF should-build-House(agent)
      call household.set-time-build-new-household(agent)
   ELSE IF (gender is male AND marital-status is married)
      THEN IF should-should-Elope(Person potential-Wife)
      returns boolean
      IF (potential-Wife is single AND NOT (Incest prohibition)
          diff ← potential-Husband-Age - potential-Wife-Age
          IF diff <= 5 THEN chance ← 100%
          Else IF diff > 5 AND diff <= 9 THEN chance ← 90%
          Else IF diff > 9 AND diff <= 15 THEN chance ← 75%
          Else IF diff > 15 AND diff <= 20 THEN chance ← 50%
          Else IF diff > 20 OR diff >= -3 THEN chance ← 20%
      END
3. call update-routine-exchange-time(current-time)
4. IF chance-remove-friend
   THEN call remove-random-friend
END

**Pseudocode: household-update(current-time)**
1. IF (no members alive in the household) OR no-adult-guardian
   call relocate-members
   EXIT
2. IF time-build-new-household equals current-time
   call build-new-household
END

**Pseudocode: should-Elope (Person potential-Wife) returns boolean**
IF (potential-Wife is single AND NOT (Incest prohibition)
    diff ← potential-Husband-Age - potential-Wife-Age
    IF diff <= 5 THEN chance ← 100%
    Else IF diff > 5 AND diff <= 9 THEN chance ← 90%
    Else IF diff > 9 AND diff <= 15 THEN chance ← 75%
    Else IF diff > 15 AND diff <= 20 THEN chance ← 50%
    Else IF diff > 20 OR diff >= -3 THEN chance ← 20%
ELSE chance ← 0
IF (chance) THEN return TRUE
ELSE return FALSE
END

Pseudocode: do-Dyadic-Exchange (current-Time-step)
1. IF (current-Time-step == ordinary-Exchange) call perform-ordinary-exchange
2. IF (current-Time-step == fiesta-Exchange) call perform-fiesta-exchange
END

Pseudocode: perform-ordinary-exchange
1. potential-recipients ← Union(list(friends), list(compadre))
2. sort(potential-recipients, dyadic-exchange-scheme)
3. num-sent-Exchanges ← Pick top 20-40% from potential-recipients
4. FOR i ← 1 to num-sent-Exchanges
   call send-Ordinary-Service (potential-recipients[i])
END

Pseudocode: perform-exchange
1. potential-recipients ← Union(list(compadre), list(friends), list(relatives))
2. sort(potential-recipients, dyadic-exchange-scheme)
3. num-sent-Exchanges ← Pick top 20-40% from potential-recipients
4. FOR i ← 1 to num-sent-Exchanges
   call send-Fiesta-Service (potential-recipients[i])
END

Pseudocode: should-Neighbour-Parent-Household (Person son)
IF similar(wife, mother) OR similar(son, list(siblings))
   THEN return TRUE
ELSE return FALSE
END

Pseudocode: determine-New-Household-Head (Person deceased-Head)
1. IF deceased-Head’s spouse is alive
   THEN new-Household-head ← spouse
   ELSE IF deceased-Head has sons
      THEN new-Household-head ← eldest-son
      ELSE new-Household-head ← eldest-offspring
   2. IF (NOT is-adult(new-Household-head))
      call THEN relocate-household-members
END

Pseudocode: reliability-Endorsement (Person person)
var: reliability-Recall-Period
IF (num-Received-Items greater-equal 10% of reliability-Recall-Period)
   THEN endorse(person, reliable)
ELSE IF (num-Received-Items less than 10% of reliability-Recall-Period)
   AND (num-Sent-Items greater-equal 10% of reliability-Recall-Period)
   THEN endorse(person, unreliable)
END
Pseudocode: reliability-Endorsement (Person person)
1. potential-Godparents-Father ← return-Best-Endorsed-Compadre
2. potential-Godparents-Mother ← return-Best-Endorsed-Compadre
3. pick-Best-Endorsed-Random-Godparents with equal probability

Figure S-1: Average percentage of selected compadre that were exclusively friends, for the six cases described in Section 5.3 of the main article.
Figure S-2: Average percentage of selected compadre that were exclusively relatives, for the six cases described in Section 5.3 of the main article.
Figure S-3: Average percentage of selected compadre that were neither friends nor relatives (miscellaneous/others), for the six cases described in Section 5.3 of the main article.
Figure S-4: UML class diagram for the project source code showing associations between key classes and their member functions.
Figure S-5: UML activity diagram for the update function in the Person class.