

Computational Social Sciences

Juan A. Barceló  
Florencia Del Castillo *Editors*

# Simulating Prehistoric and Ancient Worlds

 Springer

# Chapter 14

## Personalities, Physiology, Institutions and Genetics: Simulating Ancient Societies with Intelligent Virtual Agents

Tomas Trescak, Anton Bogdanovych and Simeon Simoff

### 14.1 Introduction

Intelligent virtual<sup>1</sup> agents are autonomous computer programs that are represented in<sup>2</sup> a virtual reality environment by human-like (or animal-like) 3-dimensional figures (called avatars) that move around the reconstructed environment and simulate its inhabitants. The use of virtual agents in cultural and historical simulations has become an important way of enriching 3D reconstructions and helping an observer not only to inspect building and artefacts, but also to understand how the reconstructed site has been enacted in the past.

With modern advancement in research and development, we are now reaching the stage when reconstructing a heritage site becomes more affordable. One possible way of reducing the development cost is to automate the design of the reconstructed virtual environment. Such design automation can be achieved with the use of design grammars—a procedural approach to generating historically informed designs of high complexity, allowing for large cities to be created in a matter of days rather than months. One of the well known examples of using this approach in historical reconstructions is the Rome Reborn project (Dylla et al. 2009), where a virtual reconstruction of the entire city of ancient Rome in the period of 320 AD was generated by automatically placing procedurally generated buildings onto a map of the

---

<sup>1</sup>See the prototype video at: [http://youtu.be/ZY\\_04YY4YRo](http://youtu.be/ZY_04YY4YRo).

<sup>2</sup>See the prototype video at: <https://www.youtube.com/watch?v=-jDsyOLZHN4>.

T. Trescak (✉) · A. Bogdanovych · S. Simoff  
University of Western Sydney, Penrith, Australia  
e-mail: t.trescak@uws.edu.au

A. Bogdanovych  
e-mail: a.bogdanovych@uws.edu.au

S. Simoff  
e-mail: s.simoff@uws.edu.au

city produced by archaeologists. However, most historical reconstructions similar to Rome Reborn do not employ virtual agents in their simulations, as the development cost for these agents is high and advanced automation techniques similar to design grammars are not yet readily available for building large populations of virtual agents.

Modern video games are a good illustration in regards to the possibilities that arise with employment of virtual agents in simulating human behaviour. Players of modern games often experience complex human-like interactions with virtual agents and agents themselves become one of the most important aspects of game play and one of the key entertainment factors. Due to problems with automating agent development the cost of developing video games is very high and it is often hard to justify such high spending in non-profit areas of research that usually require cultural and historical simulations. An example that illustrates the magnitude of spending in video games is *Crysis 3*, a popular game with the estimated cost of developing being in excess of \$66 Million (Gauder 2013). It's hard to imagine such level of spending when it comes to historical simulations, so populating a historical environment with virtual agents needs to be automated.

Aiming to achieve cost saving, some researchers model their virtual societies at the level of crowds rather than individual agents. One well known example of using "virtual crowds" in historical simulations is the visualisation of the Roman Colosseum (Gutierrez et al. 2007), where a crowd simulation approach is taken to visualise the spectators in a gladiator fight. While virtual crowds essentially consist of a large number of virtual agents, designing a crowd normally comes down to designing a few individuals and then replicating them a desired number of times with slight modifications so that the crowd appears to be diverse. The state of the art in using agent crowds in historical simulations is outlined in Mam et al. (2007) where a virtual City of Pompeii is populated with a large number of simulated people, who simply walk around the city avoiding collisions. In this work the virtual agents help to give an impression about the appearance of the ancient people who used to populate Pompeii, but these people are not involved in historically authentic interactions. So they play the role of moving decorations and can only extend the atmosphere of the culture simulation, while offering little in regards to understanding everyday life in the simulated society.

A number of crowd simulation and crowd generation approaches appear in the literature but hardly any of them advance beyond having avatars moving around and carrying objects with them. Further in the paper we show how through simulation of physiological needs and motivations together with personality traits we can achieve much more sophisticated simulations of human behaviour. Furthermore, employing genetic methods for inheriting personality traits and appearance characteristics and connecting virtual agents with formalisations of social roles and social norms allows for a similar (or even higher) level of complexity in large agent crowds as seen in commercial video games. In contrast to standard video games, however, the cost of development for such agents can be greatly reduced through a high degree of automation that our approach offers.

The remainder of the paper is structured as follows. Section 14.2 presents motivation for selecting the combination of genetics, social norms, personality and physiology as a way of advancing the state of the art in historical simulations. Section 14.3 presents our methodology to be employed for creating such simulations. Section 14.4 shows how the aforementioned methodology was applied to building a historical simulation of everyday life in the ancient city of Uruk, 3000 B.C. Section 14.5 shows the application of our methodology to an Australian cultural heritage case study, in which we simulate the life of an Aboriginal Darug tribe. In Sect. 14.6 we analyse the results obtained from the case studies. Finally, Sect. 14.7 summarises the contribution and outlines the directions of future work.

## 14.2 Approach

Simulation of life in 3D reconstructed historical cities is a costly and time-consuming process, comparable in cost and efforts to development of a commercial video game (involving years of development and millions of dollars in funding (Gauder 2013)). Costs and effort can be decreased with automatic generation of population. This is a two-fold process, in which we need to generate the unique appearance and the behaviour of each individual. Unique appearance can be generated by mimicking the biological reproduction, as for example in Trescak et al. (2012). One way of automatization of behaviour is to represent individuals as autonomous virtual agents that can generate their goals and act upon them (Vosinakis and Panayiotopoulos 2001). To generate such goals, we propose to use motivation, and in particular physiological motivation, such as hunger, thirst, fatigue and comfort. In this case agents generate their goals upon physiological trigger, e.g. getting hungry. If needed, other types of motivation can be employed, such as safety, love, or self-realisation (Maslow et al. 1970; Alderfer 1969).

The problem with classical approaches to agents driven by physiological motivation is that in a historical simulation all such agents would follow the same circadian rhythm (get hungry, thirsty at the same time), what leads to undesired, uniform behaviour. To avoid this, in our methodology, we propose to configure *motivational modifiers*, which affect the decay rate of a given motivation. For example, a hunger modifier affects the pace in which an agent gets hungry. If such modifiers are different for every agent—then every individual follows its own circadian rhythm, executing goals at various time intervals, increasing believability of the simulated population.

In classical Artificial Intelligence (AI), in order to achieve a goal each agent needs a plan. Such plans can be automatically generated using traditional planning techniques (Shehory et al. 1999; Braubach et al. 2005). Such planning techniques normally model perfectly rational behaviour, which is not always suitable for simulating humans as this results in emotionless, “robotic” behaviour. To avoid it, in our methodology, we enrich agents with personalities and emotions, which affect their decisions when creating a plan for a current goal. This approach may even lead to

emergent agent behaviour that appears to be closer to human-like reasoning. As an example, imagine a fisherman agent with no personality and emotions, that catches fish when it's hungry. The agent will fish until it succeeds, or until it dies of hunger, unless we manually specify a possible change of plans when hunger level raises to a critical value. In contrast, the same fisherman having personality and emotions may get frustrated when being hungry and unsuccessful. This agent may “decide” to stop fishing when frustration level overwhelms the rational decision for fishing and will search for alternatives to feed, such as begging or stealing food. The decision whether to beg or steal would depend on agent's personality.

In the previous example, fisherman represents a specific *social group* of the simulated population. Social groups combine certain classes of individuals that fulfill their goals in a similar way. Combining individuals into social groups allows us to define and program actions on a group level, rather than having to do this on individual level, reducing effort in defining crowd behaviour.

In human societies, it is not uncommon for members of different social groups to interact with each other and even cooperate in order to fulfill their goals. For example, imagine a fisherman who has to trade fish with a spear-maker in order to replace his broken fishing spear (see Sect. 14.4). A common technique being used in AI to facilitate the kind of interactions between different social groups as in the example above is to employ Organisation-Centred Multi-Agent System (OCMAS). The OCMAS approach is to explicitly formalise social norms of the agent population and connect those norms to the social roles, which represent different population groups. Such social norms capture rules and protocols that drive agent interactions. As a result, agents can use these norms in reasoning to create plans for their current goal. This provides agents the ability to automatically perform their actions depending on their assigned social group.

## 14.3 Methodology

Following the approach described above, we present our methodology separated into several steps that facilitate automatic generation of intelligent agent crowds, where agents generate goals depending on physiological modifiers and plan their actions depending on their personality and in accordance with social norms.

### 14.3.1 Step 1: Design the Base Population

Base population represents the initial group of agents used to generate the rest of the crowd. This population has to define the fundamental visual properties of the resulting crowd. Therefore, for each ethnic group that will be generated, there must be at least one couple of avatars, where both individuals maintain the ethnicity-specific visual traits (e.g. asian eyes), while all other non-specific features (e.g. head shape)

are varied. Following this approach, during genetic reproduction, ethnicity-specific features are carried on to the following generations (Trescak et al. 2012), while diversity within ethnicity is assured.

This process requires a significant effort, as designers have to define all avatars with distinctive appearance and a library of related textures, clothing and attachments in order to ensure high variety. In order to reduce the effort, we propose to design and use *parametric avatars* (Lewis 2000; Trescak et al. 2012), which are avatars with visual features that can be modified using parametric values. For example, parameter “height” and “body fat” would modify the corresponding parameters of avatar body. Such parameter values of an avatar form genes combined in a chromosome used to reproduce children with diverse appearance.

To better understand how the diversity is achieved—we need to explain the process of genetic reproduction. In this process, an agent’s appearance, motivational modifiers (in our case physiological modifiers), and its personality are encoded into “genes”. As a result, these three groups of genes form three chromosomes, depicted in Fig. 14.1.

During reproduction, we take two parents and combine each of the three pairs of related parent chromosomes to produce the child’s chromosome. We decide how many genes are inherited from the father and how many from the mother using a *father-mother ratio*. A *crossover* operator is responsible for combining chromosomes. Theory of genetic algorithms defines several crossover operators, i.e. split operator, but for our purposes, we define a specific *fuzzy operator*, that imitates the biological crossover using two pairs of chromosomes (Vieira et al. 2010).

**Definition 1** Given mother’s chromosome  $c^m$  consisting of genes  $c^m = g_1^m g_2^m \dots g_n^m$ , the father’s chromosome  $c^f$  consisting of genes  $c^f = g_1^f g_2^f \dots g_n^f$ , the parent *gene selector function*  $s_{f^m}^i : 2^G \rightarrow \{0, 1\}$  which for position  $i$ , where  $0 \leq i \leq n$ , selects either mother or father gene depending on probability given by the father-mother ratio  $r^{f^m}$  and the *fuzzy function*  $f : \mathbb{I} \rightarrow \mathbb{R}$  which for gene on position  $i$  selects a random value in the interval given by  $f(i) = [s(i), (g_i^m - g_i^f)/2]$ , we define a **fuzzy** crossover operator  $\odot : C \times C \rightarrow C$  as  $c^m \odot c^f = f(1) \cdot f(2) \dots f(n)$ .

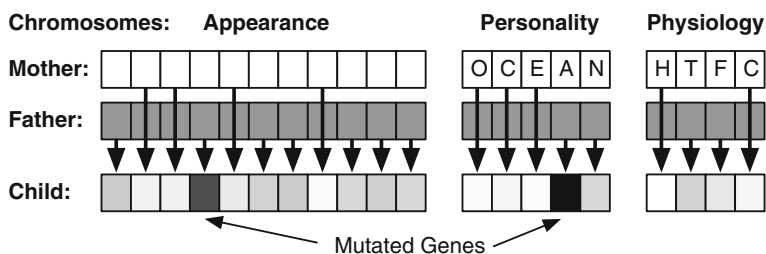
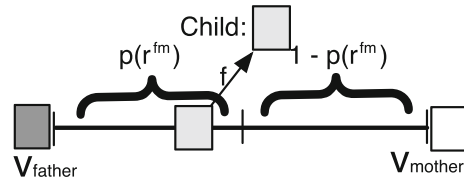


Fig. 14.1 Genetic reproduction using “Fuzzy” operator

**Fig. 14.2** Using genetic operators to form an agent's chromosome



Fuzzy operator creates a new gene value by selecting a random value from the interval defined by the gene values of the parents and depending on the specified father-mother ratio takes this value closer to father or mother gene. This process is depicted in Fig. 14.2, where  $r^{fm}$  means father-mother ratio and  $p(r^{fm})$  means probability of selecting value from the interval, depending on  $r^{fm}$ .

Another important process of the biological reproduction is mutation, which is the driving mechanism of evolution and novelty in species. We mimic the mutation process by modifying the value of pre-defined number of genes to the value from outside of the previously mentioned interval. The result of genetic manipulations is a new chromosome using which we can reconstruct a new child, its appearance, physiological needs and a personality.

Once the appearance of the avatars representing the base population has been specified in a parametric fashion—a diverse crowd of a desired size can be automatically generated following the aforementioned genetic principles. The agents in the crowd will have diverse appearance, while at the same time the important ethnic features of their appearance will be preserved. In order to introduce diversity of their behaviour—further steps of the methodology need to be completed starting with the configuration of motivational modifiers.

### 14.3.2 Step 2: Configure Motivational Modifiers

Genetic approach is also used to diversify agent behaviour. For this purpose, *motivational modifiers* are encoded into genes of the chromosome. Therefore, in this step, for each member of the base population the *motivational modifiers* are specified. In case of physiological motivation, these modifiers relate to hunger, thirst, fatigue and comfort, and represent the decay rate in which agents are getting hungry, thirsty, tired and sleepy. To avoid an impression that every single agent follows the same day cycle and performs the same set of actions at the same time, these values must be different for every agent from the base population. The more diverse these values are in the base population, the more diversity will be present in the circadian rhythms of the resulting crowd.

### 14.3.3 Step 3: Specify Personality Traits

While diverse motivational modifiers assure execution of actions at various times, agent personalities determine the kind of actions the agents will execute. In this step, for each member of the base population its personality is specified using the popular OCEAN model (Goldberg 1990), which captures five personality traits: *openness*, *conscientiousness*, *extroversion*, *agreeableness* and *neuroticism*. Openness relates to imaginative, creative aspect of a person. Conscientiousness captures the ability to be organised and careful. Extroversion defines, how social and outgoing a person is. Agreeableness relates to ability to cope with people, friendliness and generosity. Neuroticism defines tendency for negative emotions and instability.

Combination of the OCEAN values defines a specific character. Explaining, how to define a specific character is out of scope of this work, therefore we regard interested reader to existing publications (Bartneck 2002; Steunebrink et al. 2009). For the purposes of this methodology, it is important that agents forming the base population have different personality values, so that during genetic reproduction their children will have new, emerging personalities. In Sect. 14.6, we present how the diversity of parent personalities affects their children, and how it determines which actions they select as the result of having a certain personality type.

In order for agents to be able to select an action that is most relevant for their personality, such action has to be annotated by following *personality facets* (Howard and Howard 1995): *temptation*, *gregariousness*, *assertiveness*, *excitement*, *familiarity*, *straightforwardness*, *altruism*, *compliance*, *modesty* and *correctness*. Using values of personality facets, the agent selects an action that provides the highest utility for its personality type (Bartneck 2002; Howard and Howard 1995). See Table 14.1 for an example of annotations for *work*, *beg*, *steal* and *search* actions.

Often, actions such as “work” have various meaning in the context of different social groups. Working for fishermen means to catch fish, while for pot makers it means to make pots. Therefore, in the next step of the methodology, the institution is specified, which defines all the social groups, their interactions and also defines the meaning and parameters of specific actions, e.g. determines how quickly a particular object satisfies hunger.

**Table 14.1** Personality facets of agent actions

	Tempt.	Gregar.	Assert.	Excitement	Famil.	Altruism	Compliance	Modality.	Corr.
Beg	0	0	-0.5	0	0	0	0.5	0	0.5
Work	0	0	0.5	0	0	0	0	0	1
Search	0.5	0	0.75	0.5	0	-0.25	-0.5	0	-0.5
Steal	1	0	1	1	0	-1	-1	0	-0.75



### 14.3.4 Step 4: Formalise Social Norms and Roles

To define social groups, their actions and interactions, an Electronic Institutions (EI), a well established Organisation-Centred Multi-Agent System (OCMAS) is specified. EI establishes what agents are permitted and forbidden to do as well as the constraints and the consequences of their actions (Esteva 2003). In general, an EI regulates multiple, distinct, concurrent, interrelated, dialogic activities, each one involving different groups of agents playing different roles. Definition of an EI consists of the following four components:

First, a *dialogical framework* specifies social roles involved in the simulation and their hierarchy. Figure 14.3a depicts the role structure of the simulation of Uruk 3000 B.C and Fig. 14.3b depicts the one of Aboriginal simulation (see Sect. 14.6). Apart from the role structure, the dialogical framework defines ontology, a common language for communication between agents.

Second, a *performative structure* isolates specific activities (also called scenes) that can be performed within an Electronic Institution. It defines how agents can legally move among different scenes (from activity to activity) depending on their role. Furthermore, a performative structure defines when new scene executions start, and if a scene can be multiply executed at run time. A performative structure can be regarded as a graph whose nodes are both scenes and transitions (scene connectives), linked by directed arcs (See Fig. 14.4). The type of transition allows to express choice points (Or transitions) for agents to choose which target scenes to enter, or synchroni-

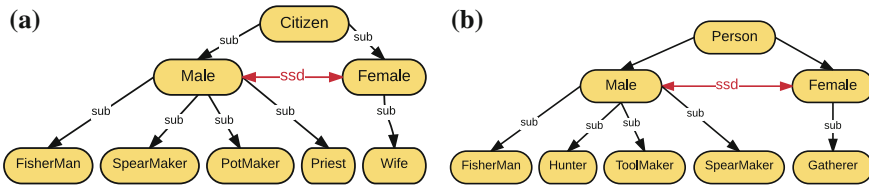


Fig. 14.3 Role hierarchy. a Uruk 3000 B.C. b Australia

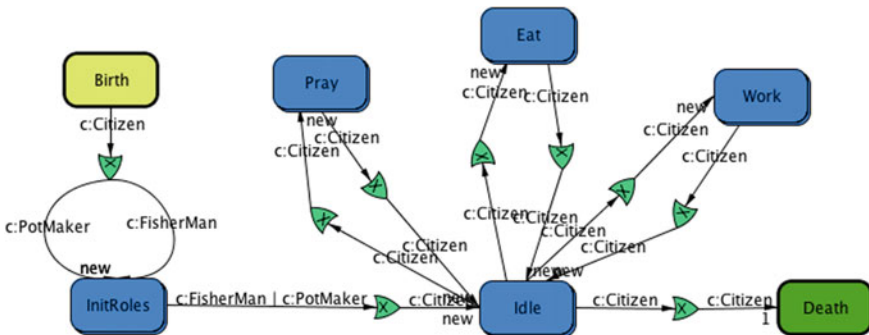


Fig. 14.4 Performative structure

sation/parallelization points (And transitions) that force agents to synchronise before progressing to different scenes in parallel. The labels on the directed arcs determine which agents, depending on their roles, can move between scenes to transitions.

Third, for each activity, interactions between agents are articulated through agent group meetings expressed as *scene protocols*, which follow well-defined interaction protocols, whose participating agents may change over time (agents may enter or leave). A scene protocol is specified by a directed graph whose nodes represent the different states of a dialogic interaction between roles (See Figs. 14.5 and 14.6). Its arcs are labelled with illocution schemes (whose sender, receiver and content may contain variables) or time-outs.

Definition of EI is fundamental to agent reasoning and our dynamic planning algorithm that constructs a list of actions to fulfill the current goal by finding a path (sequence of actions) that make the agent go into the desired scene and reach a desired state within this scene.

An institution provides agents with knowledge about possible actions that can be performed. The next step of the methodology provides means of visualising these actions in the virtual world.

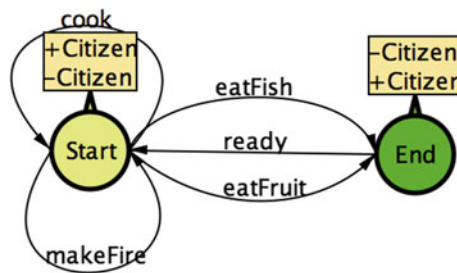


Fig. 14.5 Eat scene protocol

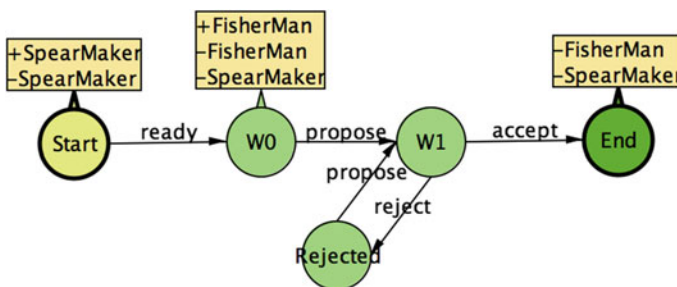


Fig. 14.6 Trade scene protocol

### ***14.3.5 Step 5: Adaptation and Annotation of the Environment***

For purposes of visualisation, institutional actions must have corresponding objects, animations and scripts. In this step, objects of the virtual world related to such actions are created and annotated with specific meta-data, so that agents know that a connection between institutional illocutions and objects is established. Agents use annotations in their planning, which is affected by the current state of the environment. Therefore, interactive objects have to contain information on what action they provide and what are the action parameters (Trescak 2012).

Adaptation and annotation of the environment is the last step that requires manual input. In this last step we generate the population of the simulation and make it act within the simulated virtual environment.

### ***14.3.6 Step 6: Generating the Population***

Generation of population is a fully automatic process, where the desired number of “children” is generated from the base population using genetic approach described in Sect. 14.3.1. Initially, children are only sets of chromosomes and their appearance has to be reconstructed in a given virtual world. Once connected to the virtual world, they start automatically generate goals and act upon them.

## **14.4 Case Study: Uruk 3000 B.C**

In order to highlight the key aspects of our approach, we have applied it to simulating one of the humanity’s first cities—the city of Uruk 3000 B.C. To further address the agility of our approach, we apply our methodology first to Second Life,<sup>3</sup> a well known virtual world platform, and then to Unity 3D,<sup>4</sup> the popular game engine. The Second Life simulation, serves to present the life of Uruk to wide public, using well-know virtual world platform with many existing users. The drawback of Second Life is in its lacking capability of handling large societies of intelligent agents (or non-playable characters, NPCs). On the other hand, Unity 3D facilitates the creation of sophisticated single-user and multi-user 3D games, and also provides the possibility to execute large societies. The size of the society is bounded only by the computational capability of the hardware, on which the game is executed. In this section, we describe how our methodology facilitates deployment of sophisticated historical simulation to both platforms and estimate and compare their work load estimates.

---

<sup>3</sup><http://secondlife.com> (last visited 06/2015).

<sup>4</sup><http://unity3d.com> (last visited 06/2015).

### 14.4.1 Preparation: Designing the World

Before we can apply our methodology, we need to design the 3D environment of the simulation. In Second Life, we started with an existing 3D model of the city that included key buildings, plants, animals and terrain. This model was developed by archaeologists and provides some level of historical exactness. For Unity 3D, we have recreated this 3D model in Google Sketchup and Blender. Furthermore, we have modelled historical objects used by various crafts belonging to the epoch. These objects include beds positioned on roofs, various chairs and tables, pots for cooking, market equipment, pottery ring, spears, and spare spear parts, fisherman boats and rows. 3D design requires a lot of effort, and the preparation step took significantly longer than design and execution of the city population. Figure 14.7 portrays the 3D design of the market, executed in Unity 3D, with several, custom designed objects. Figures 14.8 and 14.9 compare the visualisations in both, Second Life and Unity 3D.

With the static 3D design of the environment in place, we can start applying our methodology to populate this environment with autonomous agents.

### 14.4.2 Step 1: Design the Base Population

When defining the base population in Second Life, we considered only one ethnic group of Uruk citizens. Therefore, we designed only two members of the base population portrayed in Fig. 14.10a, b, using which we have generated the rest of the population.



Fig. 14.7 3D design of the Uruk market with live avatars



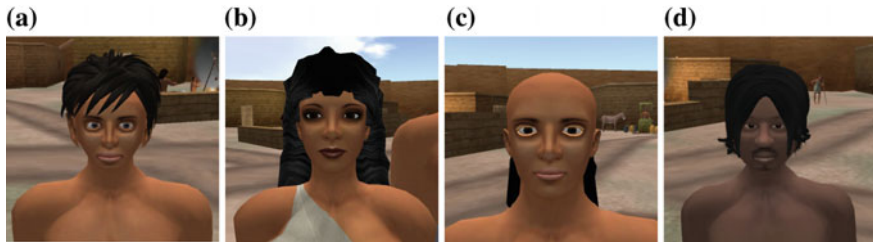
Fig. 14.8 Second Life



Fig. 14.9 Unity 3D

Figure 14.10c, depicts a child generated without mutation. This child clearly carries visual traits from both parents, having mother’s nose, but father’s mouth. Figure 14.10d depicts the child of the same parents, but with high level of mutation. Clearly, some visual traits are still visible (e.g. nose, jaw shape), yet, there are new emergent visual features, such as skin colour.





**Fig. 14.10** Generating crowd appearance in Second Life. **a** Father. **b** Mother. **c** Child. **d** Mutant

In Unity 3D, we have applied a bit different approach and we have generated the base population using the *genotype rules* (Trescak et al. 2012). Using such rules we can specify a racial or ethnic profile, which limits gene values only to the specific range. For example, we can specify what shades of skin colour can be used, what is the approximate size of the nose, what is the range of person height and so on. Yet, this approach can only generate avatars belonging to the same race/ethnic and does not allow us to generate intra ethnic avatars. Since we are generating avatars belonging to the ancient Uruk ethnic, this is not a problem.

To modify and visualise avatars in Unity 3D, we used the open source Unity Multipurpose Avatars<sup>5</sup> technology for generating random avatars. We have extended the default randomisation mechanism that has only limited control over generated avatars, with our genetic approach allowing to generate avatars belonging to a specific ethnic.

Figure 14.11 depicts the sample of ten avatars generated from the initial population of five avatars. In the base population we have two ethnics, Caucasian (Adam and Bea) and Sumerian (Cyril, Diana and Eva). We have used western names for the sumerian population only for convenience, in order to code them alphabetically by the first letter in their name (A–E). Generated children are named by coded names of their parents, the crossover operator and the mutation level used during generation process. To portrait the preservation of ethnic features we have designed all members of sumerian population with bigger, distinctive noses and darker skin colour, while caucasian population has smaller noses and and lighter skin colour. Child of C + D in the first row and C + E in the second row obviously carry on only the sumerian features, although C + D shows also very distinct features, due to the high level of mutation that has been used. Interesting result is in the second row, where we depict four different children of A + E, each of them visually distinct, yet clearly carrying features from both father and mother. Two children are quite small, with lighter skin or bigger ears as their father, others are taller, or with darker skin and smaller ears as their mother.

<sup>5</sup><http://u3d.as/content/uma-steering-group/uma-unity-multipurpose-avatar/67d> (last visited 04/2014).

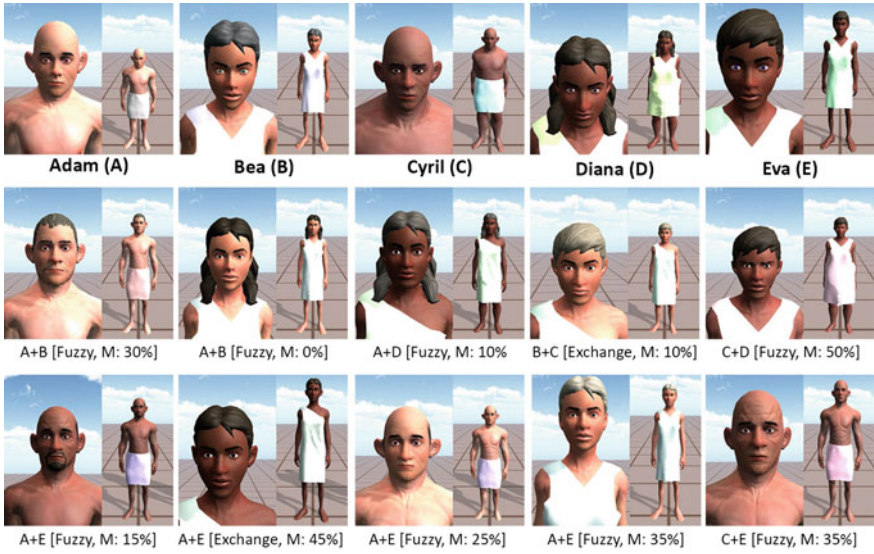


Fig. 14.11 Detail of the crowd generated for Unity 3D



Fig. 14.12 Overview of the crowd generated for Unity 3D

Figure 14.12 depicts the society of 150 avatars, generated from base population belonging to the same ethnic. As a result, none of the generated avatars carries caucasian traits and skin colour. The apparent difference is the overall graphic quality, which is prevailing in Unity 3D.

### 14.4.3 Step 2: Configure Motivational Modifiers

Base population serves not only to generate avatars with unique appearance, but also with a unique (or non-uniform) behaviour. As a result, in the next step, we defined the physiological modifiers of the base population. We set various decay rates for hunger, thirst, fatigue and comfort for each member of the population. Avatars generated from the base population will obtain varied and mutated values of these modifiers. Since each modifier will have a different value, avatars will become hungry or tired in distinct intervals, executing their actions non-uniformly. Figure 14.13 shows the graphical user interface, that facilitates the specification of physiological properties.

### 14.4.4 Step 3: Specify Personality Traits

Physiological motivation solves the (when) problem of uniformity, when agents execute their actions at various time frames. On the other hand, having avatars with distinct personalities solves the (what) problem of uniqueness, when agents perform actions matching behavioural profile. As a result, we define personalities for each agent using the OCEAN model. Figure 14.14 shows the graphical user interface, that facilitates the specification of personality properties. In Sect. 14.6 we describe the setups for personalities that were used.



Fig. 14.13 User interface for the definition of physiological properties of an avatar

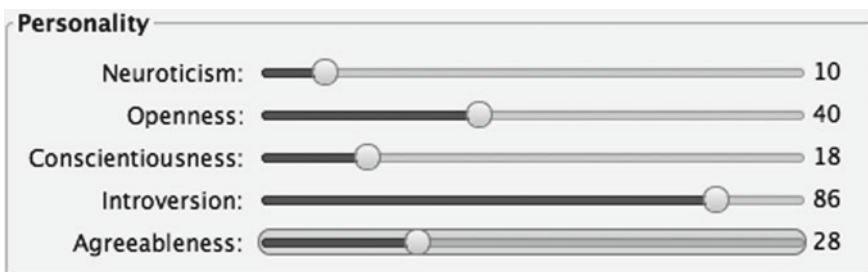


Fig. 14.14 User interface for the definition of personality properties of an avatar



Apart from the definition of agent personalities, we annotated all actions and relate them to a specific personality, using personality facets (see Sect. 14.3.3). Table 14.1 shows four actions that Uruk agents perform to satisfy the goal of “eating”. In this table, there are four actions, i.e. beg, work, search and steal, and nine personality facets, e.g. temptation, gregariousness, assertivity with valued ranging from  $-1$  (low) to  $1$  (high). These facets work as modifiers used to calculate utility of a given action in relation to a specific personality. The higher the utility, the more probable is that the action will be selected. “Stealing” action is defined for agents with more aggressive personalities (very low correctness, low altruism), “begging” for agents with low-confidence (very low assertivity, higher correctness) and “working” and “searching” for more neutral personalities with varying sense of correctness. It is probable, that we will have to adjust these values later on, but for now it sufficed.

#### 14.4.5 Step 4: Formalise Social Norms and Roles

Next, we defined all components of the Electronic Institution, with roles of fisherman, spear-maker, pot-maker, pries, king and wife (see Fig. 14.3). All of these roles are sub-roles of *citizen*, which holds all common properties for all roles, e.g. inventory of owned items.

Then, we defined possible actions of agents in specific scene protocols. For current roles we defined pray, eat, make spear, make pot, trade and fish protocol (Fig. 14.15). Make spear, make pot and fish protocol belong to the scene “Work”, and



**Fig. 14.15** Working on the land

agents select the correct protocol based on their role. Most of these protocols only command a single agent what actions need to be performed to achieve its goal. The exception is the fishing protocol, which defines collaborative actions for two agents, where one agent has to row a boat, while the other is fishing. Therefore, fisherman always have to agree to go fishing in pairs.

Finally, we grouped scene protocols in a performative structure (see Fig. 14.4), which restricts execution of actions in scenes to specific roles.

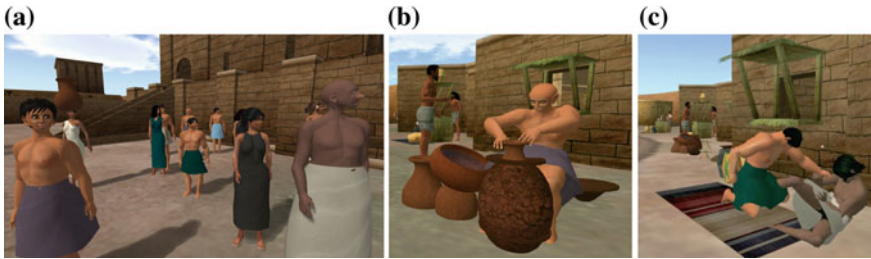
#### ***14.4.6 Step 5: Adaptation and Annotation of the Environment***

For all actions and interactions, we have recorded animations, such as begging or stealing food, using motion capture and copied them in “.BVH” format to Second Life and with the help of Blender 3D, we have converted “.BVH” file to Unity 3D. Recording animations and their subsequent processing in any platform is a very delicate task and usually requires professional crew and equipment. Since we had no such possibility our own acting performance sufficed.

Moreover, since 3D object carry no meta-information on their possible purpose, we have added related objects to the virtual world model and annotated the environment so that agents can use them in their planning. For example, agents use the 3D object camp fire to cook their food. Therefore, we annotate that this 3D object provides action (illocution) “cook” from the scene “Eat” (annotated as action: Eat.cook). As a result, when agent plans its action, it knows that it has to interact with camp fire object to perform the “Eat.cook” action. In another example, we annotated the a pottery ring with “action: Work.PotMaker.makePot”, what defines that pottery ring provides action makePot in the scene protocol “PotMaker” from the scene “Work”.

#### ***14.4.7 Step 6: Generating the Population***

In the last step, we generated a population of agents and connected these to Second Life. Each agent had a unique appearance and automatically started fulfilling its goals. Agents were correctly selecting their goals based on their physiology, executing them at various intervals due to different physiological modifiers, planing their actions based on their personality and social norms, and executing them in the simulated environment. Figure 14.16 shows some virtual agents from the resulting simulation.



**Fig. 14.16** Everyday life in the city of Uruk 3000 B.C. **a** Crowd. **b** Working. **c** Stealing

## 14.5 Case Study: Darug Clan, Australia

In our second case study, we simulate the life of an Australian Aboriginal clan from the Darug tribe, living in the area of Parramatta, New South Wales, in times before the arrival of first fleet and the establishment of the European settlement. This simulation was built in Unity 3D in the form of an educational video game and a virtual reality experience.

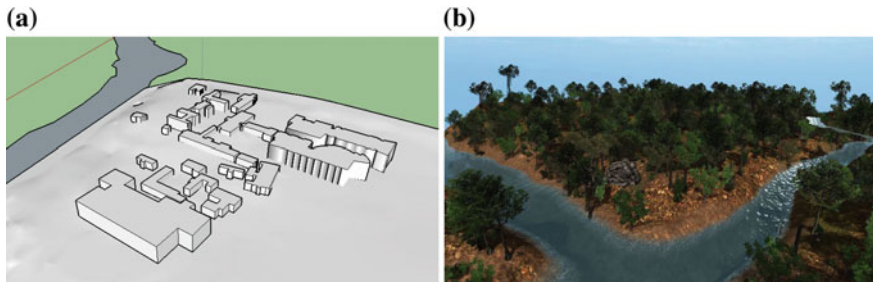
The interactive 3D video game takes the player on a quest to explore the life of an Aboriginal clan in the Parramatta basin. A spiritual mentor and the guardian in the form of an aboriginal elder gradually introduces the participant to the daily life of native clans, the knowledge they possessed, rituals they performed, protocols they kept and their connection to dreamtime. The elder familiarises the player with various clan members as they perform their daily activities such as tool making, painting, fishing or preparing food. During these interactions the player also learns about the aboriginal medicine, arts, as well as ceremonies, such as the smoking ceremony and receives an introduction to their spiritual values.

The virtual reality simulation uses Oculus Rift <sup>(6)</sup> headset to take the user on an immersive journey in historical Australia. The information provided is the same as in the video game, yet the content is not interactive, and user partakes the role of a sole observer listening to the spiritual mentor. In the remainder of this section, we describe how we applied our methodology to deliver this historical simulation and discuss the believability of our approach.

### 14.5.1 Preparation: Designing the World

The initial step of the 3D simulation creation process was the artistic design of 3D assets that formed the simulation environment. This phase highly depended on the invaluable help from the Elders of the Darug clan, who consulted us on the believability of our simulation. Elders selected the location of the simulation to correspond

<sup>6</sup><https://www.oculus.com/> (last visited (06/2015)).



**Fig. 14.17** Environment design. **a** Initial model (SketchUp). **b** Final model (Unity)

with grounds, upon which the South Parramatta campus of the University of Western Sydney is located. The initial model of the environment was constructed from the GIS data using Google Sketchup<sup>7</sup> (see Fig. 14.17a). We have modified this initial model in Unity 3D and added local flora and fauna (see Fig. 14.17b). Then, we have implemented the animal behaviour (artificial intelligence), including the flocking behaviour of kangaroos, moving lizards and snakes, flying birds and bats and fish swimming in the river. While bats and lizards follow pre-defined paths, kangaroos show intelligence by fleeing from moving humans and placing bait in the water attracts fish. We implemented the animal behaviour using popular Unity 3D plugins: Playmaker<sup>8</sup> and NodeCanvas.<sup>9</sup> Use of these plugins facilitates the reusability of the developed functionality, and its visual nature helps the non-technical team members actively participate in its specification.

With the static 3D design of the environment in place, we apply our methodology to populate the simulation environment with autonomous agents.

### 14.5.2 Step 1: Design the Base Population

The population in this simulation consists of a single ethnic group. Therefore, we designed only two members of the base population to generate the rest. While both designed avatars carry typical aboriginal features, such as the wide nose or dark hair, we have also introduced variations of visual properties, such as skin colours or height. Maintaining consistent values (or just a minimal variation) of stereotypical features in all designed avatars, significantly increases the probability of their propagation to descendants, and their exclusion can only be affected by the mutation. On the other hand, features that we wanted to vary, we assigned from both extremes of

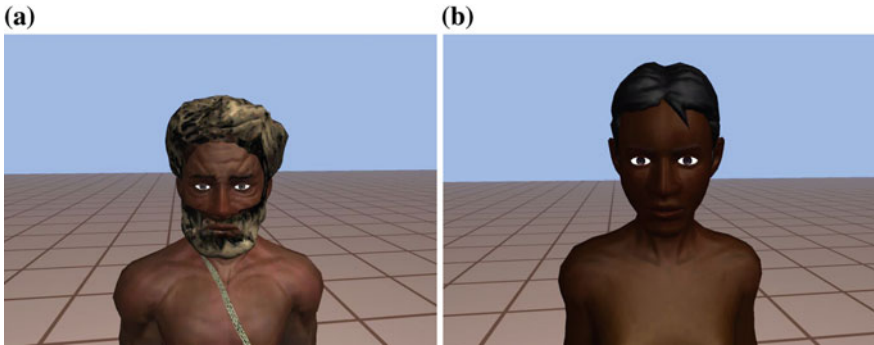
<sup>7</sup><http://www.sketchup.com>.

<sup>8</sup><http://nodecanvas.com>.

<sup>9</sup><http://www.hutongames.com>.

the feasible range. For example, we designed a strong and bulky male and thin and fragile female. Using the fuzzy operator to crossover genes, we obtain avatars with varied body and face builds (Fig. 14.18).

Consequently, consulting aboriginal elders, historic literatures and photographs, we have designed traditional clothing, tools and wearables. Since aboriginal population in the simulated era wore clothing that often did not cover the private parts, in order to target wider audience in which some may be offended by nudity, we decided to use the design of more recent clothes. Figure 14.19 depicts the generated crowd



**Fig. 14.18** Base population used to generate the Aboriginal population



**Fig. 14.19** Aboriginal crowd. **a** Father. **b** Mother

of males and females, wearing traditional hip or neck bags and variations of loin clothes from the kangaroo or possum fur. To assign random clothes to avatars, we have used the popular Unity Multi-Purpose Avatar (UMA) plugin.<sup>10</sup>

### ***14.5.3 Step 2: Configure Motivational Modifiers and Step 3: Specify Personality Traits***

Physiological motivation drives agents' proactive goal creation. Motivational modifiers affect the pace in which they get hungry, thirsty or tired. Having varied values assures that agents take decisions at various intervals. Motivational modifiers and personality traits form the part of the chromosome that is used by the genetic reproduction. Since we used only two members of the base population, similarly to visual features, to gain variation in descendants, we have assigned motivational modifier values from both sides of feasible extremes. We created a male that was getting hungry at a high pace, needs to drink often, yet takes a long time to get tired. Female is humble on resources, yet fatigues quickly.

Concerning avatars personalities, we used the very same approach, providing significantly different OCEAN profiles for both avatars, with aggressive, yet timid male and submissive, yet social female.

### ***14.5.4 Step 4: Formalise Social Norms and Roles***

In this step, we defined components of the Electronic Institution. The dialogic framework contains roles of fisherman, spear-maker, hunter, tool-maker and gatherer (see Fig. 14.3b). All of these roles are sub-roles of *person*, which holds common properties for all roles, e.g. inventory of owned items.

Please note that this role hierarchy does not correspond to the actual tribe structure. In the real world, tribesmen perform various tasks, where individuals design their spears, create own tools and go hunting with them. This knowledge is incrementally passed onto by elders. We have separated the activities between various roles to isolate effectively agent behaviour and goals. Also, currently there exist only limited means for the dynamic plan creation with a large decision space. By dynamic plan creation we mean a possibility of agents to automatically *generate and adapt* plans that lead to the fulfilment of their goal (e.g. they do not need to re-start the whole plan, but continue from the last feasible point).

During the design of the Electronic Institution, we were able to re-use a large portion of scene protocols from the Uruk project (e.g. eating, gathering, fishing), supporting the usability of our approach. The reason we can reuse EI functionality

---

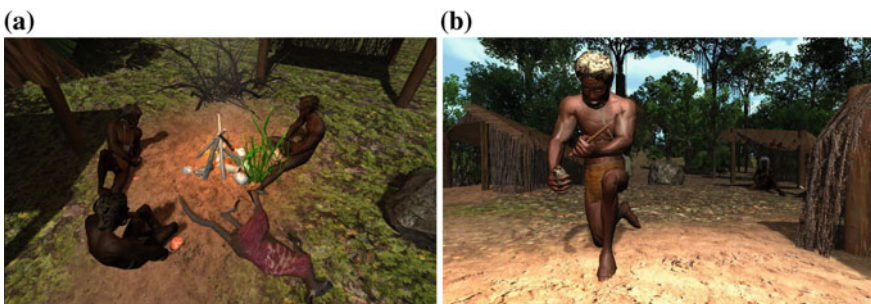
<sup>10</sup><http://uma.unity3d.com/> (last visited 06/2015).



is that protocols for action execution remained the same, what changed is their visualisation in the 3D environment. For example, in the fishing scene, one fisherman controls the boat while the other is using a spear to catch fish. The process is the same in Uruk as well as with Darug clan, but Darug fishermen no longer sit and stand in the boat, they kneel. Also, not a long paddle but a piece of bark is used to steer and row the boat. As a result, we only needed to record new animations for action execution, their control remained the same.

### 14.5.5 *Step 5: Adaptation and Annotation of the Environment*

With the invaluable help from Aboriginal elders who performed as actors, we recorded authentic animations that portray daily chores and activities of the Darug tribesmen. These activities include simple tasks such as eating, drinking, more sophisticated activities such as making tools, spears, creating artwork and also different rituals such as the smoke ceremony, dancing or feasting (Fig. 14.20). These animations trigger upon an agent acquiring a goal that requires accomplishing one or several actions. For example, to make an axe, agents need to find a viable piece of wood, stone or chipped bone. To facilitate this process, agents need to recognise objects and their functionality to consider them in their plans. Therefore, in this step, we placed annotated world objects with the representation understandable by agents. In this project, we have also introduced *dynamic annotations* that change depending on the object state. For example, alive kangaroo is annotated as “speareable”, “food source”, but when it is killed its annotation changes to “leather”, “bone”, “food”. Agents use these annotations to create plans for accomplishing their goals.



**Fig. 14.20** Simulated activities. **a** Feasting. **b** Tool making



**Fig. 14.21** Everyday life in the Darug clan

### ***14.5.6 Step 6: Generating the Population***

In the last step of our methodology, we generate the population of agents. Using our approach, the appearance, physiological and personality profile are unique with each generated agent.

For each created agent, a Playmaker script assigns a role and initialises the connection with the Electronic Institution. Agents start the random walk to discover their environment. After an arbitrary period, physiology triggers a feeding request. To feed, agents dynamically create plans based on their role, personality and surrounding objects. For example, the only way for spear-makers to feed is to exchange their produced spears for food with fishermen or hunters. In order to create a spear, they have to use knives, which serve them to work the spear-wood. Another role, Painters, first search for ochre, water and bark and then exchange their pieces with hunters and fishermen. Figure 14.21 shows a scene from the aboriginal simulation showing tool maker and painter at work, as well as men feasting on a kangaroo.

## **14.6 Evaluation**

In this section we analyse our results and estimate the effort (in hours) needed to set-up and execute the simulations for both case studies. Then, we describe two experiments, that evaluate the diversity of generated agent behaviour.



### ***14.6.1 Uruk Simulated in Second Life***

We estimate that the total time spent on completing the case study from Sect. 14.4 was close to 7 days. The process was relatively fast as we have already had a model of the city and we focused only on generating the population. Step 1, definition of base population took us three days, where most of this time was spent on modelling clothing and attachments for avatars. Second Life provides parametric avatars with possibility to change more than 200 visual features. Therefore designing the body of the avatars took us only a couple hours per avatar. Steps 2 and 3, in our case took only one hour to complete, as the physiological modifiers and personality were defined only focusing on having wide range of values (rather than trying to achieve some pre-determined global personality skew in the resulting population). Step 4, definition of institution took 1 day, during which we designed all scenes and a performative structure and tested agent interactions. Also, we studied how to set-up the personality facets of personality-based actions. Step 5 took a lot of effort and time, in total 4 days. During this time we recorded and tuned all the animations, designed all interactive objects (e.g. pot-making ring) and scripted their behaviour. Step 6 is fully automatic, generation of 100 agents took only a few seconds, visualisation of each avatar in Second Life takes about 30 s per avatar.

### ***14.6.2 Everyday Life of the Darug People Simulated in Unity 3D***

We estimate that the total time spent on completing the case study from Sect. 14.5 was close to 31 days. In contrast to the Uruk simulation, the Darug design has not been provided to use, so we needed to spend significantly more time on the initial design. The increase in time is due to the fact, that we needed to re-create manually the 3D design of the terrain and houses in the reconstructed area (7 days) as well as all 3D objects (5 days) and avatar clothing (6 days). Three days were spent on recording animations. Converted animations had to be adjusted and programmed to be used with Unity (i.e. Mecanim). The conversion and animation adjustments took us 2 days. Then, we have annotated the environment with meta-data used by agents during reasoning about possible plans to accomplish their goals. In this case annotation is done directly in Unity 3D, via custom MonoBehaviour objects. With the help of the aboriginal elders we have then designed a simple institution, personality setups for the base population, their daily plans and related cultural information for the institutional roles. As a result it took us only 1 day, to adjust steps 2–6 to Unity 3D.

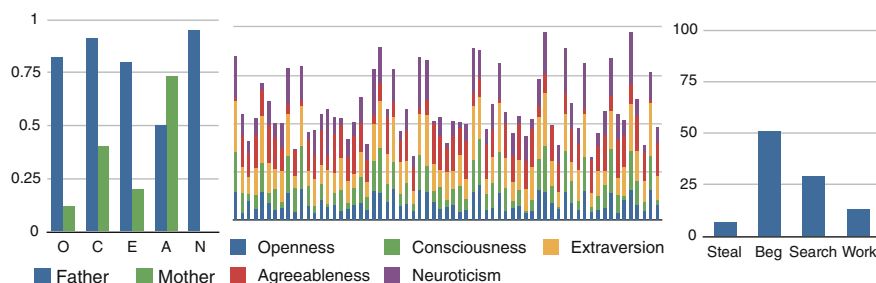
### 14.6.3 General Methodology

Using our methodology, in combination with modern game engines and 3D virtual worlds, we significantly cut down the time to populate historical 3D simulations. The drawback of our approach is that we rely on parametric avatars with ability to modify the avatar appearance and clothing using declarative (visual) parameters. But, this is not a major issue, since we already possess the technology for Unity 3D and Second Life, and other game engines offer similar functionality, although in the form of paid plugins.

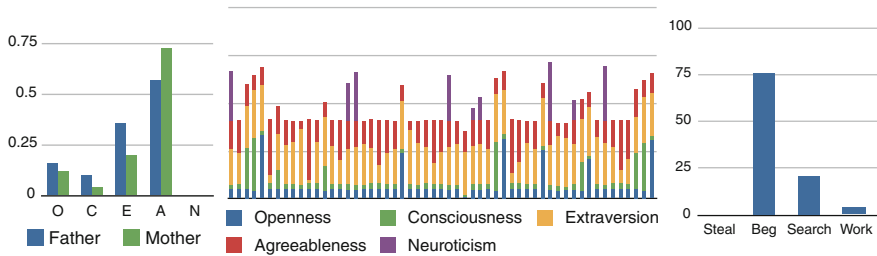
Having parametric avatars and employing our genetic approach we can generate unique, ethnic avatars in a very little time. Using motion capture, we can easily animate these avatars and believable results depend only on exact historical data and acting skills. Furthermore, using the Electronic Institution technology, we can declaratively specify the social structures and interaction protocols, used by agents to automatically reason about their possible actions. Electronic institution can be tweaked during the simulation runtime, decreasing the debugging efforts in comparison to traditional approach, where simulation has to be restarted after every change.

### 14.6.4 Generating Children of Parents with Diverse Personalities

To test the validity of generating agents with various behaviour, we performed two experiments. In the first experiment, we set-up diverse personalities of parents, where one parent had very low confidence, while the other was very aggressive (see Fig. 14.22a). When hungry, one parent chooses to beg, the other one to steal. Then, we have generated their 100 children, with father-mother ratio set to 30% (agents will have 70% of their genes closer to their mother). Figure 14.22b depicts the highly varying personality profiles of their children. We let generated agents decide what to



**Fig. 14.22** Experiment 1: Children of parents with opposite personalities (no mutation). **a** Parents. **b** Children personalities. **c** Actions



**Fig. 14.23** Experiment 2: Children of parents with similar personalities (mutation 25 %). **a** Parents. **b** Children personalities. **c** Actions

do when hungry and observed emerging behaviour of searching for food and working in 40 % of generated children (see Fig. 14.22c). Only a few children decided to steal as the father-mother ratio was in favour of the mother.

### 14.6.5 Generating Children of Parents with Similar Personalities

In the second experiment we set-up father and mother with similar personalities (see Fig. 14.23a) and during generation applied a high level of mutation (25 %). We observed the children personalities and actions, depicted in Fig. 14.23b. Generated children had very similar personalities, with occasional exceptions, due to mutations. In this experiment father chooses to search for food, while mother chooses again to beg. Having the same father-mother ratio (30 %), most of children decide to beg, just like their mother (see Fig. 14.23c). Several mutated children decided to work.

The above experiments showed that having a base population with diverse personalities leads to generating children with diverse behaviour. Having parents with similar personalities results in their children having similar personalities and predominantly showing the same behaviour, unless they undergo mutation.

## 14.7 Conclusions

In this work, we have presented a methodology for generating large and diverse agent populations for the purposes of social simulations. This methodology is using genetic operations to produce individuals with unique appearance and behaviour. We have separated the methodology into six steps. First step is the definition of the base population, which specifies the visual traits of the whole population, although using mutation we may achieve novelty during generation. Second step is the definition of motivational modifiers, where motivation serves as the goal selection mechanism. In

our case, we used physiological needs as the main motivation. Third step is the definition of personality traits, where personality affects agents decisions during planning and agents select actions that best match their profile. In the context of social simulations, agents belong to specific social, ethnic or cultural groups and have to obey specific social norms. Therefore, fourth step is the definition of the social system and norms, in our case using Electronic Institutions. The fifth step is the adaptation and annotation of the environment that reflects all actions specified in the electronic institution. Agents are using these annotation to automatically plan their actions and interact with the environment. Following these steps results generating a diverse agent population having a high degree of variety in their appearance and behaviour, while also demonstrating substantially high degree of complexity of actions being performed by the agents.

We have illustrated the application of the methodology proposed in this paper to the development of two case studies. In the first case study virtual agents were used to enrich a historical reconstruction with simulation of everyday life of ancient Sumerians in the city of Uruk, 3000 B.C. In the second case study we have applied our methodology to building a cultural simulation of the Darug tribe in Australia around 1700 A.D. Due to the high degree of automation in the creation of large virtual agent groups that our methodology offers, in both case studies we were able to achieve significant time savings while maintaining a high degree of complexity of the resulting virtual agent behaviour.

## References

- Alderfer CP (1969) An empirical test of a new theory of human needs. *Organ Behav Hum Perform* 4(2):142–175
- Bartneck C (2002) Integrating the occ model of emotions in embodied characters. In: *Workshop on virtual conversational characters*. Citeseer
- Braubach L, Pokahr A, Moldt D, Lamersdorf W (2005) Goal representation for bdi agent systems. In: *Programming multi-agent systems*. Springer, pp 44–65
- Dylla K, Mueller P, Ulmer A, Haegle S, Fisher b (2009) Rome Reborn 2.0: a case study of virtual city reconstruction using procedural modeling techniques. In: *37th Proceedings of the CAA conference*. Archaeopress, Oxford, pp 62–66
- Esteva M (2003) *Electronic institutions: from specification to development*. Ph.D. dissertation, Institut d'Investigació en Intel·ligència Artificial (IIIA), Spain
- Gauder J (2013) *Crysis 3 cost \$66 million to make, can next gen sustain such budgets?* GameChup Video Games News at <http://www.gamechup.com/crysis-3-cost-66-million-to-make-can-next-gen-sustain-such-budgets/>, 1 Mar 2013
- Goldberg LR (1990) An alternative “description of personality”: the Big-Five factor structure. *J Personal Soc Psychol* 59(6):1216–1229. <http://psycinfo.apa.org/doi/index.cfm?fuseaction=showUIDAbstract&uid=1991-09869-001>
- Gutierrez D, Frischer B, Cerezo E, Gomez A, Seron F (2007) AI and virtual crowds: populating the Colosseum. *Cult Heritage* 8(2):176–185
- Howard PJ, Howard JM (1995) *The big five quickstart: an introduction to the five-factor model of personality for human resource professionals*

- Lewis M (2000) Evolving human figure geometry. Ohio State University, Tech. Rep. OSU-ACCAD-5/00-TR1, ACCAD
- Mam J, Haegler S, Yersin B, Muller P, Thalmann D, Van Gool L (2007) Populating Ancient Pompeii with crowds of virtual Romans. In: 8th international symposium on Virtual Reality, Archeology and Cultural Heritage—VAST
- Maslow AH, Frager R, Fadiman J (1970) Motivation and personality, vol 2. Harper & Row, New York
- Shehory O, Kraus S, Yadgar O (1999) Emergent cooperative goal-satisfaction in large-scale automated-agent systems. *Artif Intell* 110(1):1–55
- Steunebrink BR, Dastani M, Meyer J-JC (2009) The OCC model revisited. In: Reichardt D (ed) *Emotion and Computing*. Paderborn, Germany
- Trescak T (2012) Intelligent generation and control of interactive virtual worlds. Ph.D. dissertation, Autonomous University of Barcelona
- Trescak T, Bogdanovych A, Simoff S, Rodriguez I (2012) Generating diverse ethnic groups with genetic algorithms. In: Proceedings of the 18th ACM symposium on virtual reality software and technology, ser. VRST '12. ACM, New York, NY, USA, pp 1–8. <http://doi.acm.org/10.1145/2407336.2407338>
- Vieira RCC, Vidal CA, Neto JBC (2010) Simulation of genetic inheritance in the generation of virtual characters. In: VR, Lok B, Klinker G, Nakatsu R (eds) *IEEE*, pp 119–126
- Vosinakis S, Panayiotopoulos T (2001) Simhuman: a platform for real-time virtual agents with planning capabilities. In: Proceedings of the IVA 2001 Workshop. Springer, pp 210–223