City of Uruk 3000 B.C. : Using genetic algorithms, dynamic planning and crowd simulation to re-enact everyday life of ancient Sumerians

Tomas Trescak
School of Computing,
Engineering and Mathematics
University of Western Sydney,
Sydney, Australia,
t.trescak@uws.edu.au

Anton Bogdanovych
School of Computing,
Engineering and Mathematics
University of Western Sydney,
Sydney, Australia,
a.bogdanovych@uws.edu.au

Simeon Simoff
School of Computing,
Engineering and Mathematics
University of Western Sydney,
Sydney, Australia,
s.simoff@uws.edu.au

***Abstract*—Virtual reality reconstructions of ancient historical sites have become a valuable technique for popularising science and visualising expert knowledge to general audiences. Most such reconstructions only re-create buildings and artefacts and place them in the context of the virtual environment, but what is often missing in such simulations is the ability to see how ancient people lived their daily life in these environments. Our presented case study shows how the use of genetic algorithms and simulation of physiological needs helped us to populate the 3D reconstruction of the city of Uruk with a large crowd of intelligent agents simulating daily life of ancient Sumerians in Uruk[1].**

I. Introduction

Uruk was an ancient mesopotamian city located in the present day Iraq. It is believed to be one of the first human built cities on Earth. In our reconstruction we simulate Uruk in the period around 3000 B.C and employ Unity 3D[2] as the engine for visualising Uruk.

One of the key challenges of this work was how to have the virtual Uruk populated by virtual agents that re-enact everyday lives of its ancient inhabitants and how to do it with maximal possible automation and cost saving. In the following sections we present the key details of our approach built around genetic algorithms, crowd simulation, artificial physiology and dynamic planning.

II. Obtaining expert data

In order to populate the city with virtual agents, we designed a number of scenarios that we obtained after detailed discussions with subject matter experts and history consultants. As the result of these discussions we identified roles the agents play, scenes they participate in, interaction protocols and social norms. We followed the methodology described in [1] to structure the knowledge received from the experts and transform it into formalisations suitable for developing the underlying multiagent system.

[1] See the video at: http://youtu.be/ZY_04YY4YRo
[2] http://unity3d.com

The agents in the Uruk simulation represent a slice of Uruk society among which are fishermen families, priest, king and a number of workers (i.e. pot maker, spear maker). The agents can sense changes in the environment state, which result in them updating their beliefs accordingly. They are supplied with a number of internal goals and plans to reach those goals. The current implementation features fishermen families where men's daily routines include sleeping, eating, fishing and chatting. The females do house work, sleep, eat bring water from the well and go to markets. The king agent walks around his palace and invites students to ask him about his ruling strategies. The priest agent conducts a prayer in the temple, accepts gifts and explores the city. Other agents represent various workers: pot makers, spear makers, etc. Those workers produce goods, exchange goods with one another, attend the prayer and in their spare time explore the city, provide information to students and simulate social interactions with other agents.

III. Implementation Details

The 3D design of the city was created based on the results of archaeological excavations and available written sources. The objects and artefacts were mainly created based on the details obtained from museums, but several objects were also created following drawings and illustrations from history books. Figure 1 shows Uruk reconstructed in Unity.

One of the key problems with the Uruk simulation was to populate the city with virtual agents simulating daily life of its citizens (Sumerians). The scenarios mentioned in the previous section if implemented in a classical way (where every agent is individually designed and programmed) would be extremely time-consuming and costly to produce. Therefore, we have developed an approach to automate many steps. The automation lies in the technique to automatically generate a crowd of avatars of a desired size using a genetic algorithms approach from a small initial sample of manually designed avatars representing the base population. In order to make these avatars perform complex daily life routines the agents are supplied

Fig. 1: 3D Visualisation of the city of Uruk 3000 BC

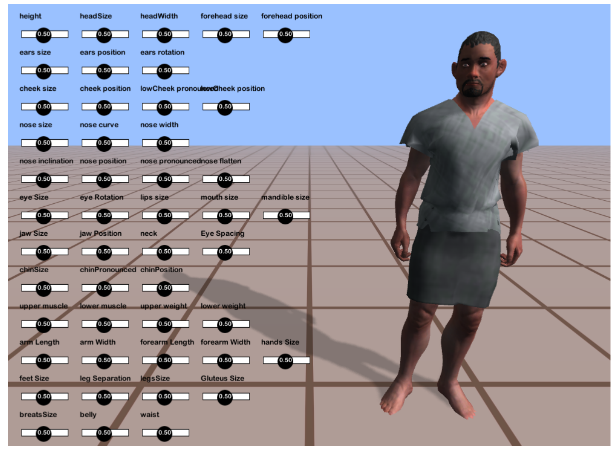

Fig. 2: Unity Multipurpouse Avatar - Modification

with models of physiological needs, so that agent goals can be automatically generated through these needs (like hunger, thirst, fatigue, etc.) becoming more pronounced. The agents must have a way to react to these needs in a way appropriate for the social role the agent is playing in the reconstructed society. Thus, to make this possible we formalise the role flow and social norms of the reconstructed society (that we label as institution). The institution permits agents to find a plan of actions that leads to satisfying each of the needs, while keeping this plan in accordance with the role played by the agent. Finally, to increase believability and improve diversity of agent actions we supply agents with diverse personalities, so that actions of others and the state of the environment may affect an agent's emotional state and result plan variations in response to the same goal. Further we highlight the technical details of the aforementioned techniques. This process is separated into six steps, following the methodology from [2].

a) Step 1: Define Base population: The process begins by defining the base population of the city of Uruk. According to our methodology, the base population has to include at least one pair of avatars, male and female, for each ethnicity living in the city. To define the base population, we propose to use parametric avatars, which is a mesh that can be modified using pre-defined parameters, e.g. height, head shape, eye size. Parametric avatars are well known from computer games, where they form a part of the closed eko-system, not usable for our purpouses. Therefore, we selected the open-source project, Unity Multipurpouse Avatars (UMA)[3], which allows to modify avatar shape directly in Unity 3D (see Figure 2). While UMA permits to modify the avatar body shape, the Marvellous Designer[4] is used to reconstruct avatar clothing, accurately according to the available literature. We then use Blender to create various attachments and trinkets that enhance avatar appearance. UMA uses these objects and distributes them randomly during the generation process (see Section III-A).

[3] http://fernandoribeirogames.wix.com/umabeta
[4] http://www.marvelousdesigner.com

b) Step 2: Configure motivational modifiers: Our aim is to generate avatars not only with unique appearance, but also with a unique (or non-uniform) behaviour. Therefore, in this step, we define the physiological modifiers of the base population. We set various decay rates for hunger, thirst, fatigue and comfort for each member of the population. As a result of this process, avatars generated from the base population obtains varied and mutated values of these modifiers. Since each modifier has a different value, avatars become hungry or tired in distinct intervals, executing their actions non-uniformly. In order to facilitate this in Unity 3D, we designed the component which contains editor of physiological modifiers, monitors the current physiological state of an agent, as well as allows to edit these values at runtime. Figure 3 depicts the interface of the physiology component and the agent drinking water as a reaction to passing the threshold value for thirst.

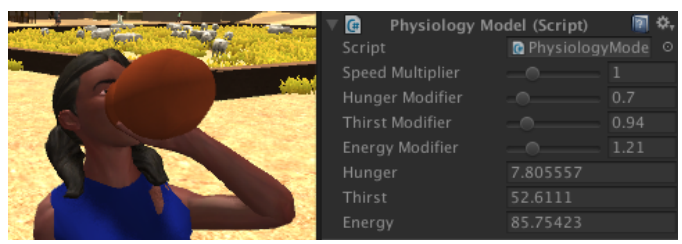

Fig. 3: Physiological needs: agent reacting to state "thirsty"

c) Step 3: Specify personality traits: When agents obtain several possibilities of reaching their goal (e.g. to steal, beg or work to obtain food), we propose them to select the action which best reflects their personality and a current emotional state. Therefore, in this step, for each member of the base population its personality is specified using the popular OCEAN model [3], which captures five personality traits: *openness, conscientiousness, extroversion, agreeableness and neuroticism*. Similar to physiology, we have developed Unity components that facilitate the definition of agent personalities and monitor their current emotional state.

d) Step 4: Formalise Social Norms and Roles: In this step, we formalise the social structure and corresponding actions and rituals of the simulated society [4]. For this task, we specify an Electronic Institution (EI), a well established Organisation-Centred Multi-Agent System (OCMAS). EI establishes what agents are permitted and forbidden to do as well as the constraints and the consequences of their actions [5]. Definition of an EI consists of the following components: dialogical framework (defines the common ontology and roles), performative structure (defines role access), scene protocols (define interactions) and norms (control interactions). The definition stored in the Electronic Institution is used by agents to detect the structure of interactions needed to fulfil their current goal (see Section III-B).

In case of Uruk, we defined roles for Fisherman and Potmaker as well as their corresponding actions in scene protocols, such as fishing and pot making. Also, actions to satisfy physiological needs, such as drinking, eating or resting form part of the institution. Visual counterpart to institutional actions is defined by Behaviour Trees in Unity 3D (see Section III-C).

e) Step 5: Adaptation and Annotation of the Environment: In the previous steps, we have introduced means for automatic goal selection based on physiological needs and defined the institution that facilitates automated plan making based on infrastructure of interactions. When agents want to execute such generated plan, they need to interact with other avatars and objects in the virtual space. In case of objects, they have to be annotated in order for agents to understand their functionality. This annotation includes actions that can be performed with given object (e.g. apple has action "eat", fish has actions "catch" or "cook"), constraints of their use (e.g. fish has action "cook" only if if is owned by agent) and also consequences of their use (e.g. "eating" fish decreases the level of hunger more than eating an apple).

Apart from the annotation of objects with actions they provide, we need to annotate actions with the emotional response that is triggered when performing the action. This drives agent decision to select an action that is most relevant for their personality; such action has to be annotated by following *personality facets* [6]: *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 [7] [6]. See Table I for an example of anotations for *work*, *beg*, *steal* and *search* actions. "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.

A. Generating the Population

In the last step, we encode avatar's visual, physiological and personality parameters values into genes, which form strings of genes or chromosomes that identify each avatar. Then, using approaches from genetic algorithms, by combining chromosomes from two parents we reproduce the rest of the population automatically, where each new avatar is unique. Figure 4 depict a generated crowd of 100 agents in Unity3D.

Fig. 4: Generated Crowd in Unity 3D

B. Dynamic Planning and Prioritisation

In our approach, rather that giving agents full "recipes" on how to accomplish a specific goal, we give them only ingredients (in the form of annotated actions) and means of combining them to fulfil their goals. These means come in form of dynamic planning algorithm and prioritisation based on the actual goal. First, we take a look at our prioritisation mechanism, then we explain the dynamic planning for the Uruk simulation.

In our approach, we use agent sensors to trigger desires to fulfil a goal. Currently, we have two sensors: physiological sensor, which detects hunger, thirst and energy, and the scheduling sensor, which feeds agent information about its schedule. For example, agents wake up around 7AM and go to work around 8AM. Each sensor feeds desires with a different priority, where physiological sensor has the highest priority. In case that agent is currently performing an action with lower priority, it pauses its execution, processes action with higher priority and then resumes the original action. If action with the same priority arrives, it is stored in the priority queue and executed after the current action. Using this approach we were able to model agent behaviour, when agent feed when hungry, drink when thirsty, or they drop to their knees whenever Uruk king is passing by (high priority action).

Prioritisation decides what goal is currently planned to execute, but dynamic planning transforms this goal into plan of actions. Our dynamic planing solution relies on environment annotation. The virtual environment contains a number of objects that can potentially be used by virtual agents and those objects can be acted upon. Through text annotations, those object specific actions are associate with pre-condition and post-conditions. So, those annotations define how an agent is potentially able to achieve its goal through atomic actions, given all possible states [2].

	Temptation	Gregariousness	Assertivity	Excitment	Familiarity	Altruism	Compliance	Modality	Correctness
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

TABLE I: Personality facets of agent actions.

C. Visual Representation of Actions in Unity 3D

While dynamic planning tells agents "what" has to be done to fulfil a goal, agents do not know "how" to accomplish it in virtual space. Agent possibly knows, what objects it has to interact with, yet it does not know how to operate them and how it will be visualised. For example, when agent decides to eat and apple, first, an apple has to be attached to its hand, and then apple eating animation is played. Eating fish is a completely different story, where agent first has to cook the fish, then put it on a plate, then sit down and eat the fish. The number of actions depends on the level of required complexity.

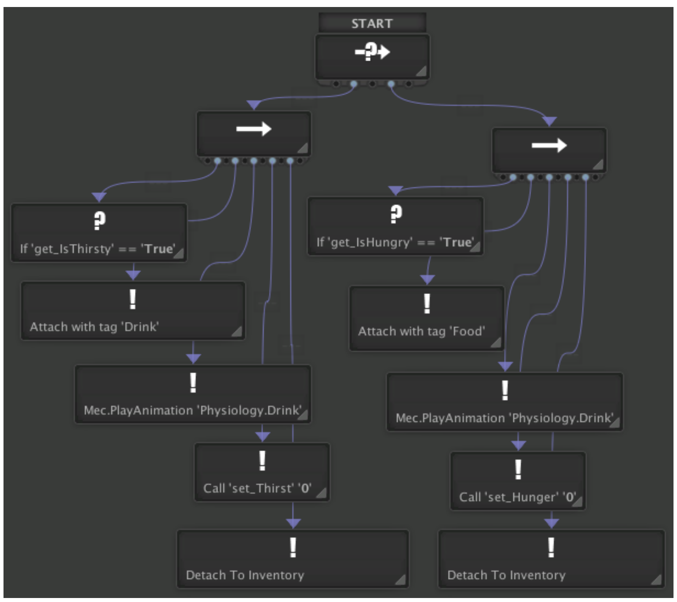

Fig. 5: Excerpt From The Behaviour Tree

To facilitate the definition of visual behaviour of avatars and objects we use behaviour trees and final state machines, provided by the NodeCanvas[5] plugin. Figure 5 depicts an excerpt from the behaviour tree for the actions performed for fulfilling certain physiological desires, such as drinking or eating an apple. Same approach is taken for interactive objects, which also use behaviour trees to drive actions of participants that operate them. For example a chair owns a behaviour tree that instructs participant how it can sit on this chair (different chairs can trigger different sitting animations).

D. Human in the Loop and Avatar Interactions

While previous sections deal only with population of agents, we would like to also discuss the human participation in the simulation and means of interacting with the Uruk simulation. In order to best educate simulation participants we have decided to create a simple plot, which human participant can follow in order to best discover the city, its history and the sumerian culture occupying it. To follow this plot, human has to interact with agents and object to listen to their stories. The dialogue trees have been implemented using the NodeCanvas plugin, which also visualises dialogues in the game. Figure 6 depicts an interaction with one of the agents. It also shows the simple game interface with the mini-map and several GUI elements showing the progress in the game.

[5] http://nodecanvas.com

Fig. 6: Streets of Uruk

REFERENCES

[1] A. Bogdanovych, J. Rodriguez, S. Simoff, A. Cohen, and C. Sierra, "Developing Virtual Heritage Applications as Normative Multiagent Systems," in *AOSE '09*. Springer, 2009.

[2] T. Trescak, A. Bogdanovych, and S. Simoff, "Populating virtual cities with diverse physiology driven crowds of intelligent agents," in *Proceedings of the 2014 Social Simulation Conference*, 2014. [Online]. Available: http://doi.acm.org/10.1145/2407336.2407338

[3] L. R. Goldberg, "An alternative "description of personality": The Big-Five factor structure," *Journal of Personality and Social Psychology*, vol. 59, no. 6, pp. 1216–1229, Dec. 1990. [Online]. Available: http://psycinfo.apa.org/doi/index.cfm?fuseaction=showUIDAbstract&uid=1991-09869-001

[4] A. Bogdanovych, J. A. Rodriguez-Aguilar, S. Simoff, and A. Cohen, "Authentic interactive reenactment of cultural heritage with 3d virtual worlds and artificial intelligence," *Appl. Artif. Intell.*, vol. 24, pp. 617–647, July 2010.

[5] M. Esteva, "Electronic institutions: From specification to development," Ph.D. dissertation, Institut d'Investigació en Intelligència Artificial (IIIA), Spain, 2003.

[6] P. J. Howard and J. M. Howard, "The big five quickstart: An introduction to the five-factor model of personality for human resource professionals." 1995.

[7] C. Bartneck, "Integrating the occ model of emotions in embodied characters," in *Workshop on Virtual Conversational Characters*. Citeseer, 2002.