



**HAL**  
open science

## Needs Model for an Autonomous Agent during Long-term Simulations

Lysa Gramoli, Jérémy Lacoche, Anthony Foulonneau, Valérie Gouranton,  
Bruno Arnaldi

► **To cite this version:**

Lysa Gramoli, Jérémy Lacoche, Anthony Foulonneau, Valérie Gouranton, Bruno Arnaldi. Needs Model for an Autonomous Agent during Long-term Simulations. AIVR 2021 - 4th IEEE International Conference on Artificial Intelligence and Virtual Reality, Nov 2021, Taichung, Taiwan. pp.1-5. hal-03396091

**HAL Id: hal-03396091**

**<https://hal.science/hal-03396091v1>**

Submitted on 22 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Needs Model for an Autonomous Agent during Long-term Simulations

Lysa Gramoli  
Orange, Rennes, France  
lysa.gramoli@orange.com

Jérémy Lacoche  
Orange, Rennes, France  
jeremy.lacoche@orange.com

Anthony Foulonneau  
Orange, Rennes, France  
anthony.foulonneau@orange.com

Valérie Gouranton  
Univ Rennes, INSA Rennes, Inria,  
CNRS, Irisa, France  
valerie.gouranton@irisa.fr

Bruno Arnaldi  
Univ Rennes, INSA Rennes, Inria,  
CNRS, Irisa, France  
bruno.arnaldi@irisa.fr

**Abstract**—Simulating human behavior in virtual environments is still a challenge. Yet, this feature is crucial to make them more believable. Therefore, many approaches have been proposed in order to find ways to faithfully simulate human behavior. One of the key features to make a virtual agent more believable is the simulation of needs such as hunger or tiredness. Unfortunately, most of the existing approaches in needs simulation do not really address the issue of long-term simulations where some problems may appear such as time drift. Yet, this kind of simulation is useful in many fields like video games or virtual data generation. This is why, in this paper, we focus on the creation of a needs model designed for long-term simulations. According to this model, needs can evolve over several simulated days without interruption. This model is configured to obtain a proper relation between control and autonomy in order to have a coherent behavior during long periods. This paper deals with the key features to set up this needs model and introduces some preliminary results to check the coherence of the agent behavior.

**Keywords**—agent behavior, needs model, long-term simulation

## I. INTRODUCTION

Simulating the behavior of virtual agents is a crucial part to have a believable virtual environment. Among the main parts of the human decision process, simulating needs is essential to reach this credibility since it lets the agents managing their goal in an autonomous way [2]. However, most agent behavior models, including needs model, do not really address the issue of long periods where time drift and unusual activities may happen. Consequently, credible agent behaviors are often missing for long-term simulations. Yet, this kind of simulation is useful in many fields like archaeology [3], [11], data generation [1], or video games.

Therefore, in this paper, we focus on the creation of a needs model adapted for long-term simulations. In this model, these simulated needs evolve according to the time of day during several simulated days. We also use a level hierarchy based on Maslow's Pyramid theory [9] to prioritize needs according to their significance. Thus, using these parameters, an agent can select the most urgent needs and execute their associated activities to satisfy them. To illustrate

this process, we use a smart house simulator developed with *Unity Engine*<sup>1</sup> where an agent can reproduce daily activities in a smart house and trigger simulated connected sensors and actuators. Recent solutions, like OpenSHS [1], have demonstrated the interest of generating smart house data in a virtual environment. Indeed, as an example, such data could replace or complete real data to train algorithms for detecting or predicting daily activities [4]. Here, the explanations given by Avradinis et al. [2] are used to distinguish credibility over realism. In our case, credibility assessment is chosen over realism assessment, since our goal is not reaching a reproduction of the human brain regarding needs. Actually, we seek for an approach creating a believable behavior from the human stand-point. Such an approach would fit the requirements of the previously mentioned use cases.

The paper is organized as follow. Firstly, in section 2, we discuss existing work on needs management. Then, in section 3, an overview of our needs model is proposed. In section 4, some results are shown to identify the influence of needs parameters in the decision process. Finally, we introduce different future work to improve this model and to integrate it in a complete behavioral model.

## II. RELATED WORK

Many approaches have been made to simulate needs for agents living in different places such as cities [3], [7], [11] or houses [2], [5], [8]. Some of them, like MAGE [2] are based on Maslow's Pyramid of needs [9]. In this theory, recognized by the psychological field, needs are ordered according to their urgency level (see Fig. 1). In the case of MAGE [2], Maslow's Pyramid is used to set up needs, but few needs have been developed and the needs levels are not diversified. Regarding other papers, a threshold system is used to assess the intensity level of needs in order to know which needs must be satisfied in priority [5], [8]. De Sevin et al. [5] propose a selection system inspired by the viability zone concept [10] where 3 zones have been

<sup>1</sup>*Unity Engine*, official website: <https://unity.com>

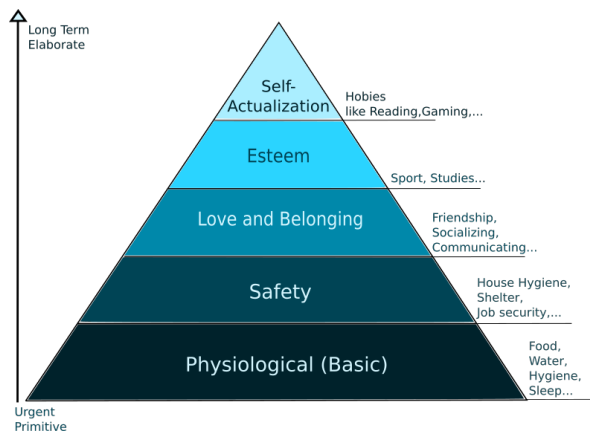


Figure 1. Representation of the Maslow's Pyramid of needs.

implemented to estimate the urgency of the needs (comfort, tolerance and danger zones). The agent goal is to maintain its needs in the comfort zone by executing the appropriate actions. Regarding the paper of Lee et al. [8], needs are more impacted by the time of day than the previous approaches. However, in these three papers, few clues are given to understand the relationship between time and needs. Thus, time drifts may appear after a certain period of time.

In the Mesopotamian population simulation [3], a comparison between the utility-based approach (used by *The Sims*<sup>2</sup> game) and a Goal-Oriented Action Planning (GOAP) approach is made in the case of needs management. In this approach, each need evolves through interaction with objects satisfying them, and a mood value (sum of the needs) is updated to choose actions. In contrast, the proposed GOAP approach uses goal-based systems to manage needs. Both approaches are interesting because they are adapted for long-term simulations. However, in the utility-based approach, behaviors seem to be scripted and scalability issues appear when a new object is added [3]. Moreover, few hints about needs management are presented because of its commercial nature. Regarding the GOAP approach, the relationship between the time of day and needs intensity is not clearly established.

Some machine learning-based approaches [6], [7] have also simulated needs. For instance, the PSI model [6] is a reinforcement learning approach where needs are simulated using liquids levels representing satisfaction levels. When the tank is not full, there is a need to satisfy. PSI structure works as a reward system: the more urgent the need is, the stronger the unpleasant signal is. The systems learn through these signals in order to avoid situations involving displeasure. However, PSI has not been tested for long periods. In another learning-based approach, a double DQN structure taking into account the Maslow's Pyramid of needs is used

to set up the agent behavior [7]. In this paper, real data is used to build the relationship between the time of day and the agent needs. Unfortunately, even though this approach has interesting results, only one day has been simulated, insufficient to know if this approach could manage some activities or needs which do not happen every day.

Judging from the previous research works, needs can be managed in various ways. However, most of them have limitations regarding long-term simulations or have not been demonstrated in this context. Indeed, the results of these papers last at most one day, which is insufficient to simulate needs evolving over several days or to simulate activities which are not performed every day.

### III. MODEL OVERVIEW

In this section, we explain our needs model specialized for long-term simulations. We based our model on Maslow's Pyramid of needs [9]. This pyramid proposes a hierarchy to distinguish the physiological needs, which are simple but imperative to satisfy in order to survive, from more elaborated needs, which are more complex but less urgent to satisfy. Contrary to previous work, our model implements several levels of hierarchy to show their impact on the choice of the needs. In addition, our model takes into account the time of day where the intensity of the needs can evolve through several days. In addition, this model can simulate occasional activities, such as sport. A certain level of non-determinism can be produced by our model. Actually, using the same input parameters, the time spent to satisfy a need can change when several possibilities can satisfy it.

#### A. Intensity of Needs Assessment

Our model uses a scoring system to select needs. Each need has a priority level calculated according to its hierarchical level and its intensity. Using this, the need with the highest priority value is selected each time another need is satisfied. Regarding the intensity of the needs, its value depends on time. Two parameters called *startTime* and *endTime* have been created to control needs progression. *startTime* is the time where the intensity of the regarded need starts to increase, whereas *endTime* is the time where the intensity of this need is at 1, the maximum value. Initially, *startTime* corresponds to the time where the simulation has started. Each need has its proper time slots [*startTime*, *endTime*], allowing us to control the duration in which needs can evolve. They are also used to constrain these needs to give us a good degree of control over needs progression. These time slots can be specific hours or a period depending on the *startTime*. Specific hours are used to constrain needs to only progress in the specified interval. Periods depending on *startTime* are used to increase needs during them and when the satisfaction is reached, *startTime* is updated in order to restart the intensity progression. These configuration

<sup>2</sup>The Sims, official website: <https://www.ea.com/fr-fr/games/the-sims>

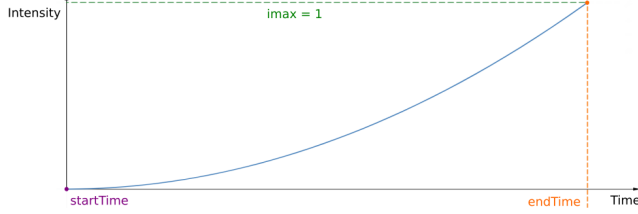


Figure 2. Need intensity according to time

are used to keep a good degree of control during long-term simulations and to avoid time drifts. When a need is satisfied, its intensity value falls to 0 and its `startTime` and `endTime` are updated. More precisely, either the next time slot is chosen when specific hours are used, either the `startTime` is updated with the time of need satisfaction when periods depending on `startTime` are used. Using this time slot method, some needs, like sport, can reach their maximum beyond 24 hours.

The shape of the intensity curve has been inspired by the needs model of *The Sims* game [3] and by the work of De Sevin et al [5]. In the first paper, most of the curves have a parabola shape evolving between  $[-100, 100]$ . In the second paper, a similar shape, based on an exponential function, has been used to assess needs intensity. In our case, a square function was used rather than an exponential function in order to avoid a too quick ascension of the intensity during the last minutes (see Fig. 2). The intensity value is limited between 0 and 1 and the curve is delimited by `startTime` and `endTime`. When the intensity reaches its maximum value, it stays at this value until the regarded need is satisfied. When this last is satisfied, its intensity immediately falls to 0. The equation of the intensity of each need  $i$  that depends on the current time  $t$  is given below:

$$i(t) = \left( \frac{t - \text{startTime}}{\text{endTime} - \text{startTime}} \right)^2 \quad (1)$$

### B. Priority calculating

After assessing the intensity of the needs, their priority is calculated between  $[-1, 1]$ . To do this, the intensity value,

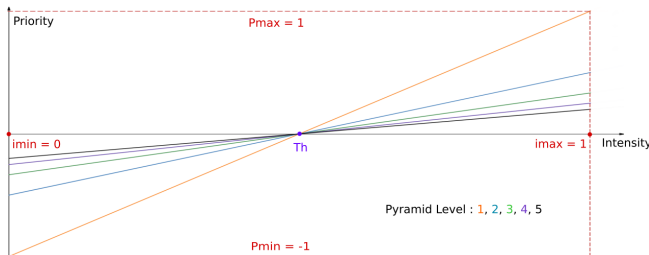


Figure 3. Needs priority according to their intensity and the Maslow's Pyrmamid Level



Figure 4. On the left, the Simulator. On the right, the agent moving in it.

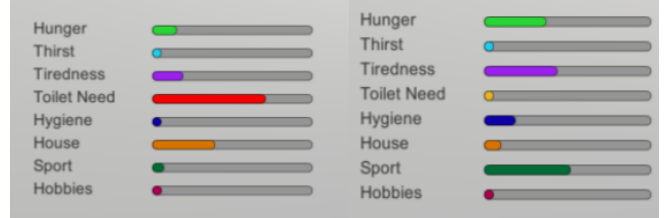


Figure 5. Needs gauges showing the evolution of the needs intensity level

the threshold value, as well as the level in the pyramid are used. The threshold is used to point out when the need is taken into account by the agent. Thus, when the intensity of the concerned need exceeds this threshold, its priority level becomes positive (see Fig. 3). This priority linearly increases in comparison to the intensity. In order to take into account the hierarchy level, a value between 1 (for the base of the pyramid) to 5 (for the top of the pyramid) is used. The equation of the priority  $P_{\text{Need}}$ , involving the intensity  $i$ , the pyramid level  $\text{Pyra}_{\text{Level}}$ , and the threshold  $\text{Th}$  is given below:

$$P_{\text{Need}} = \frac{i(t) - \text{Th}}{\text{Pyra}_{\text{Level}} \cdot (1 - \text{Th})} \quad (2)$$

### C. Need Selection and Execution

After calculating the priority value of each need, they are stored in descending order inside a list if their value is positive. Then, the first need of the list is selected by the agent. After this, the sequence of actions associated with this need is chosen and executed. When a sequence of actions is completed, a signal is sent to indicate that the need is satisfied. Before selecting another need, the priority of all needs is recalculated to know what needs are the most urgent now. If no need is selected, a default activity is chosen. Furthermore, our model is extensible since other modules can be added to influence needs and the functions calculating needs parameters can be modified.

## IV. RESULTS

To demonstrate our model, a smart house simulator was implemented with *Unity Engine* (see Fig. 4). In this simulator, the agent executes predefined actions after choosing the most accurate activity judging from the priorities of

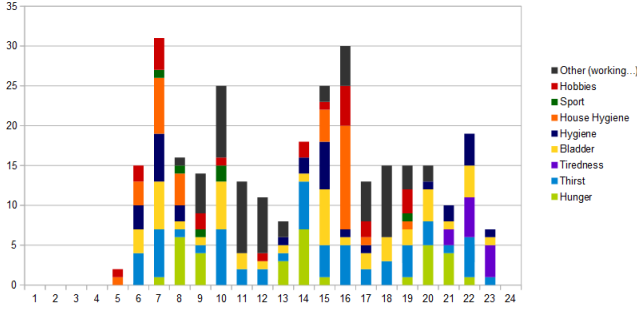


Figure 6. Frequency of activities satisfying needs during 10 days according to the hour of day

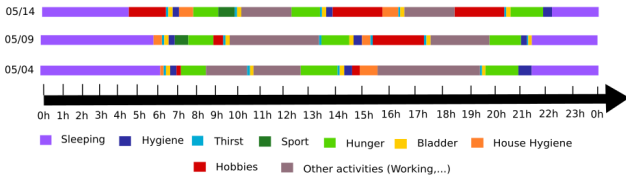


Figure 7. Timeline of three day samples showing all activities made by the agent during the daytime

its needs. Data regarding the performed activities and the associated time slots are retrieved. In addition, to have visual feedback, needs are simulated by gauges evolving over time according to the level of need intensity (see Fig. 5). When the regarded need exceeds its threshold, the gauge turns red.

The simulation was launched for 10 simulated days in which the activities made by the agent were recorded. 2 real hours are needed to produce 10 simulated days. Our simulator can fast forward time and skip periods when the agent is inactive. Table I shows the value of the needs initial parameters. The time slots as well as the average frequency observed for each activity satisfying a need are shown. Hunger need, indicated in the first row, shows an example of time slots using specific hours. The second row, indicating thirst need, uses periods depending on startTime as time slot. Thus, thirst intensity increases during 3 hours from the startTime, which is updated each time the thirst is satisfied. Therefore, our model can manage needs evolving over several days. In the future, these values will be adjusted by using real data to be closer to reality. Regarding the average of observed frequency, it directly depends on the time slot values as well as the need level inspired by Maslow's Pyramid. For instance, for the same time slot, activities satisfying hobbies appear less frequently than hygiene activities because hobbies need has a highest level in the pyramid than hygiene need. In parallel, for the same hierarchical level, activities satisfying thirst happen more frequently than hygiene, because the time slots of the thirst are smaller than the hygiene ones.

The experience uses the time slots indicated in the Table I

Table I  
ACTIVITIES FREQUENCY SATYSFYING THEIR NEED ACCORDING TO THE NEED HIERARCHICAL LEVEL AND THEIR TIME SLOTS

Need Name	Level	Time Slot	Frequency
Hunger	1	[7 a.m., 10 a.m.] [12 p.m., 2 p.m.] [7 p.m., 9 p.m.]	3/day
Thirst	1	[startTime, starTime + 3 hours]	5.8/day
Sleepiness	1	[10 p.m., 2 a.m.]	1/day
Hygiene	1	[startTime, starTime + 8 hours]	3/day
Toilets	1	[startTime, starTime + 3 hours]	5.2/day
House Hygiene	2	[startTime, starTime + 10 hours]	3.5/day
Sport	4	[startTime, starTime + 48 hours]	0.6/day
Hobbies	5	[startTime, starTime + 8 hours]	2.5/day

to control the needs intensity progression. These results are shown in Fig. 6 and Fig. 7. The Fig. 6 shows the activities made by the agent during 10 days according to the time of day. Judging from this figure, a routine seems to appear, since the agent eats and sleeps at regular times. Regarding the Fig. 7, a timeline of 3 remote days is shown, illustrating activities made during the daytime. This timeline shows that time variations exist between the days but the hours where a similar activity is carried out remain close, proving that data does not drift over time. Thus, both figures illustrate that time slots are effective to control the needs progression.

These results show that our need model can simulate needs evolving over several days. Thus, some activities which are not made every day can be simulated. Some parameters like time slots are used to improve the control over the evolution of the needs, limiting drift effects which sometimes happens in long-term simulations. Thus, the agent behavior remains coherent during long periods.

## V. CONCLUSION AND FUTURE WORK

In conclusion, we presented a needs model to manage the agent behavior during long-term simulations. With this model, needs have different levels of urgency, based on Maslow's pyramid [9], and can be simulated during several days in a controlled way. Some parameters can be changed to adapt to the priority of each need. The results show that our model can produce coherent behaviors even after a long period. However, additional works are needed to demonstrate the behaviors credibility. Real data or observations from real users could be used in the future to validate this.

This paper focuses on needs simulation but other cognition systems could be added to improve the agent credibility. In the future, we would like to set up an interaction modeling between some needs (e.g., sport would influence tiredness and so on) and assess the costs of satisfying a need. We also aim to turn this needs model into a complete cognition model which is able to manage the agent preferences, more advanced time constraints for some activities that could be configured by the user, and the management of unexpected events caused by the virtual environment. After this, our goal would be to extend this model for multi-agent systems.

## REFERENCES

- [1] N. Alshammari, T. Alshammari, M. Sedky, J. Champion, and C. Bauer, "Openshs: Open smart home simulator." *Sensors*, 17(5), 1003, 2017.
- [2] N. Avradinis, T. Panayiotopoulos, G. Anastassakis, "Behavior believability in virtual worlds: agents acting when they need to", *SpringerPlus* Vol.2, p. 246, 2013.
- [3] A. Bogdanovych, and T. Trescak, "To plan or not to plan: lessons learned from building large scale social simulations." In *international conference on Intelligent Virtual Agents*, pp. 53-62, Springer, Cham, August 2017.
- [4] J. Cumin, G. Lefebvre, F. Ramparany, J. L. Crowley, "Human Activity Recognition Using Place-Based Decision Fusion in Smart Homes." *CONTEXT 2017*, Paris, France, pp.137 - 150, June 2017.
- [5] E. De Sevin, and D. Thalmann, "A motivational model of action selection for virtual humans." In *International 2005 Computer Graphics*, pp. 213-220, IEEE, June 2005.
- [6] D. Dörner, and D.C. Güss, "PSI: A computational architecture of cognition, motivation, and emotion." *Review of General Psychology*, V.17, pp. 297-317, 2013.
- [7] H. Jang, S. Hao, P. M. Chu, P. K. Sharma, Y. Sung, K. Cho, "Deep Q-network-based multi-criteria decision-making framework for virtual simulation environment." *Neural Computing and Applications*, 1-15. 2020
- [8] W. Lee , S. Cho, P. Chu, H. Vu, S. Helal, W. Song, and al. "Automatic agent generation for IoT-based smart house simulator." *Neurocomputing*, 209, 14-24. 2016.
- [9] A.H. Maslow, "A theory of human motivation", *Psychological Review*, Vol. 50, No. 4, pp.370–396, 1943.
- [10] J. Meyer, "Artificial life and the animat approach to artificial intelligence", in *Artificial Intelligence*, Academic Press, 1995.
- [11] S. Vosinakis, and N. Avradinis, "Virtual Agora: representation of an ancient Greek Agora in virtual worlds using biologically-inspired motivational agents.", *Mediterranean Archaeology and Archaeometry*, 2016.