la VIDA: A System for Value and Identity Driven Autonomous Agent Behavior in Virtual World Scenarios – Dissertation Abstract

Ursula Addison

supervisor: John-Thones Amenyo
Graduate Center
City University of New York

Abstract

There are a great variety of systems that control agents using models of human behavior. However, often times agent action is still a reflection of the system designer's expectations and desired outcome. But, how would agents behave if they had an identity similar to a human? What goals would be formed and how would those goals be realized as actions? We would like to produce agent behavior using these questions as guidance for our work. To this end we investigate how long-term autonomy is influenced by an agent's identity and how these findings can be used to direct the behavior of artificial agents. For our system *la VIDA*, first we will create a model for human identity and ultimately integrate it with a Goal-Driven Autonomy (GDA) system at the drive level.

Introduction

There are a wide range of approaches and frameworks used to generate behavior for autonomous agents. One such approach is called Goal-Driven Autonomy (GDA). GDA systems use drives, which are more abstract than goals and encapsulate a general purpose or direction for the agent's actions (Muñoz-Avila 2018). Many GDA systems are developed to address highly specialized problems and thus simply formulate and manage goals from pre-assigned drives. We are interested in GDA systems from the perspective of agents that are autonomous and develop their drives internally from experiences, memories, impressions, and so forth. In short, our interests lie with agents that are intrinsically motivated to fulfill goals according to an inner sense of purpose. For this research we will design and build an identity profile, i.e. a representation of a mental identity taking inspiration from the human psyche. We will also investigate the connection between long-term autonomy and identity in humans. Our findings will be implemented in the **Value and Identity Driven Autonomy** system, i.e. *la VIDA*, to generate the behavior of artificial agents.

Motivation

Our work has a wide application scope. To test our system, *la VIDA*, we have developed the use-case which involves an interactive virtual reality (VR) scenario between a non-player character (NPC) and human agent. The NPC's behavior should be consistent with its identity profile and aid in the scenario's immersive experience. However, it is not difficult to imagine other situations in which our research could be useful. It might be used for generating agent behavior in simulations for video games, space exploration mission critical training, development and preparations in medicine, and education and virtual workplaces. Other applications include modeling the goals and actions of agents with a particular identity profile.

Current Behavior Approaches

There are a myriad of techniques for generating agent behavior, much more than could be covered in this short review. This summary will only produce a non-exhaustive list of broad categories in which many techniques fall into. These categories include: goal-directed, behavior trees, automaton, domain-independent planners, and hybrid systems.

An agent whose behavior is goal-directed is one who determines the goals it would like to achieve and the state of the world where those goals will be true (Muñoz-Avila 2018). The motivation to bring the world to a specific state is the drive of the agent. An agent may have a certain overarching goal that can be decomposed into to many simpler goals. These simpler goals may be fixed, but in a dynamic world new goals will need to be generated and achieved over time. Examples of this system type include the Goal Generation Management framework (GGM) (Hanheide 2010) and Goal oriented action planning (GOAP) (Orkin 2004).

A behavior tree (BT) is a data structure which arose from the gaming community's need for structures that were intuitive and robust (Iovino et al. 2020). BTs provide a direct and simple way of describing actions that a agent is capable of doing. The structure is a graph with compound tasks that decompose into smaller task subtrees (Winter, Hayes, and Colvin 2010). Task execution is processed bottom-up, completing component tasks to achieve the compound task. The Component Reasoner (Dill 2011) and an evolutionary behavior tree system (Nicolau et al. 2017) fall into this category.

Finite State Machines (FSM) are one of the earlier techniques used to control agent behavior and are ubiquitous in video game development. To execute an action and transition to a new state, an FSM must know its current state, this is accomplished by continually monitoring the environment. FSM are very useful for their highly reactive nature and the degree of control they give designers over the action and state space (Antimirov 1996). An environment may potentially have an infinite number of states, but a finite subset of those states are encoded as nodes of the FSM. When one of those states is recognized, a finite set of actions may be performed when in that state. Examples are classic deterministic FSM and a system using behavioral programming with probabilistic automata and personality by Chittaro et al. (Chittaro and Serra 2004).

Planning is a popular technique for controlling agents, particularly characters in video games. Most techniques used by game developers are some variation on search planners. The planners will normally be paired with a heuristic function that can incorporate some important consideration, e.g. plan construction time, as a way of ordering plans from least to most desirable (Wilkins 1984). A wide variety of STRIPS style and HTN planners have successfully been applied to agent control.

The last category is reserved for techniques that don't fit neatly into the above groups. One such system is the hierarchical task network and behavior tree hybrid planner developed by Neufeld et al. (Neufeld, Mostaghim, and Brand 2018). This planner combines the reactivity of behavior trees (BT) with the long-term strategizing of hierarchical task networks (HTN).

Research Goals

This paper describes early conceptual work, and our thesis is to implement preliminary work for *la VIDA*. Following is a discussion of our current research goals.

1. Can *la VIDA* use its input to formulate non-trivial, decomposable, relevant, and achievable goals?

This research goal describes the requirement that the goals generated are both non-trivial and relevant within context and a sequence of such goals can lead to the scenario being completed. For example, if one of the agent's drives is to "stay alive", it could be a correct but trivial goal to remain standing in a safe starting location. The goals should be achievable in that it is possible to plan for and execute them in the given environment. Finally, goals should be abstracted i.e. they shouldn't be terminal actions such as those sequenced by the planner.

2. If *la VIDA* generates a behavior sequence from a *la VIDA* formulated goal, can an agent autonomously execute the behavior sequence to achieve the goal?

This research goal is related to the first one, but emphasizes that goals should not be strictly theoretical; there must be a problem definition and environment for which the goals can actually be executed action by action. In addition, this action sequence should be achievable by the agent without any external interference.

3. Is the behavior sequence when executed by an agent realistic and consistent with its identity profile?

The final research question relates to the quality of the behavior sequence. Assuming it is valid and achievable, is it also believable? Meaning, is it consistent with norms and expectations that humans have regarding behavior resulting from certain beliefs, roles, and personalities? This research goal is not considering the correctness of the action sequence, but instead how it might be perceived by other agents and if those perceptions are in line with the agent's identity profile.

System Overview

la VIDA is made up of a collection of sub-systems, its chief functionality is to produce behavior sequences for a single agent via goal reasoning and planning in response to its identity profile and environment. *la VIDA* departs from other autonomy frameworks in that it focuses on the agent's mental and personality profile opposed to arbitrary drives specified by a system operator.

The first phase of processing involves the goal formulation module where drives and core goals are identified. The next phase involves interleaved planning and goal management. This planning includes strategic planning and also planning for each of the abstracted goals. The final phase determines how the terminal goals will be performed in the environment, executes those goals, and returns state information to *la VIDA*. Processing is not necessarily sequential and phases can be revisited as necessary. Following is a brief explanation of each system component. Figure 1 shows the basic components and how they interact with each other.

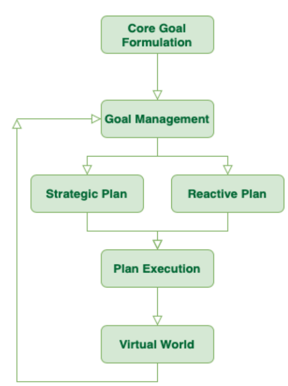

Figure 1: *la VIDA* Overview

Core Goal Formulation From a structured scenario and agent description *la VIDA* will formulate one or more core goals. Where a core goal is related to fulfilling an *integral drive* of the agent. As an example, if an agent is working for an airline in the new flight reservations department as a booking agent, a core goal could be to help each customer reserve new flights while behaving in a professional manner. Using the description which will include a role for the agent with *la VIDA*'s commonsense knowledge base, one or more core goals will be identified.

Goal Management While core goals should change very little, because they are integrally linked to the agent's mental and personality profile, subgoals may change often. Subgoals may change as a result of feedback from the environment or replanning. It is the goal management component's job to decompose core goals into subgoals and order those subgoals for the planner. Subgoals may be eliminated, added, deactivated or activated; each subgoal is given a priority value that also is subject to modification. The subgoals are eventually sent to the planner to be assigned a solution sequence. The solution sequence is an ordered list of actions that can transform the initial state into the goal state.

Strategic Plan The strategic plan also called a long-term plan is responsible for achieving goals over a sequence of actions. It is capable of making plans for subgoals that will be achieved in the somewhat distant future. Typically strategic plans aren't concerned with minute details and instead are are more abstracted than a reactive plan.

Reactive Plan The reactive plan is made of low-level actions assigned by the planner as the agent interacts in the world. By low-level actions we mean actions that can not be further decomposed and may directly be executed. These actions exist within the context of the strategic plan.

Augmenting Plans with Personality Traits and Emotions The agent has a mental and personality profile which is taken into account in the formulation of core goals and the reactive plan. Actions of the reactive plan may be augmented with tags that impact which animation, voice, or facial expression is ultimately selected when the action is being executed. We classify emotions according to the Ekman model of emotions which include: happy, sad, angry, fearful, disgusted, and surprised (Ekman 1993).

Common Sense Knowledge Base *la VIDA* will have access to a knowledge base similar to the commonsense knowledge graph (CSKG) ATOMIC-LIGHT knowledge base (Ammanabrolu et al. 2021). This knowledge base is a vast collection of common sense information and non-specialty knowledge that many humans may have. This will allow *la VIDA* to put the structured scenario and agent description into context. For instance, if the knowledge for a role is absent from the knowledge base, then *la VIDA* may not be able to formulate a core goal, and ultimately will not be able to generate any agent behavior. We may consider integrating functionality to add to the knowledge base or otherwise extending *la VIDA* dependence on a CSKG for this research or future work.

Virtual World Feedback The scenario is unfolding in real-time, as are the actions of the agents within it. To take actions that are effective and realistic, the feedback from the scenario must be used to update the world state information. This information is used by the strategic planner as it's plan is executed. For instance, if at some point the world state information deviates too much from the expected state, replanning may be necessary. The reactive planner is online and will constantly use scenario feedback in combination with the strategic plan to decide its own action sequence.

Agent The agent acts independently in the scenario, i.e. it's behavior sequence will be created via single agent planning. The agent will be able to interact with and respond to the environment and agents within it, but its actions will not require coordination. The agent will have one or more goals that can be decomposed into two or more top level goals. The agent's top level subgoals should be independent; those subgoals will need to be further decomposed so they can be planned for.

Identity Profile An identity profile is a data structure implemented as a container holding elements of type value, role, and personality. It is a summary of the most salient components of an agent's identity that are relevant to goal formulation. We will go into further detail about the identity profile in a system paper.

Usage

To use *la VIDA*, some assumptions, its input, and its output should be understood. The subsequent sections touch on each of these topics.

Preconditions

1. The structured scenario and agent description input has sufficient information for *la VIDA* to formulate a core goal.

2. A parent framework or environment manager creates and maintains the scenario returning accurate world state information to *la VIDA*.

3. Input has agent identity profile, scenario keywords, and planning problem definition.

Input We will determine the input structure and representation as our system is developed. This representation should minimally include partitioned elements for the *Agent* and *Scenario*, figure 2 is an example of one approach for structuring the input and the type of data it should hold. Some element types should only have a single instance such as *Agent*, *Scenario*, and *Role*; in future work we may increase the possible number of roles an agent may have. The other elements can have multiple occurrences, each with their row number within the parent element appended to their type. The agent may have multiple personality traits that constitute it's psychological profile, as well as multiple values that make up its mental behavior paradigm. Similarly, Scenario is expected to require multiple elements of each child type to sufficiently capture its theme and context.

```xml
<?xml version="1.0"?>
<Input>
    <Agent>
        <Role>data</Role>
        <Value0>data</Value0>
        <Value1>data</Value1>
        <Personality0>data</Personality0>
        <Personality1>data</Personality1>
    </Agent>
    <Scenario>
        <Theme0>data</Theme0>
        <Theme1>data</Theme1>
        <Context0>data</Context0>
        <Context0>data</Context1>
    </Scenario>
</Input>
```

Figure 2: Sample *la VIDA* XML input file

Output *la VIDA* output is a plan that consists of two subplans, one is the strategic plan and the other is the reactive plan. Both plans are a sequence of actions, where any action may be augmented with realism tags. The strategic plan is at a much higher level of abstraction than the reactive plan, and each action must be further decomposed before it can be executed. Each element of the reactive plan is a terminal action and may be executed directly. The strategic plan is typically generated offline, and modified online in the case of replanning. The reactive plan is generated fully online as the agent exists and interacts within the scenario.

References

Ammanabrolu, P.; Urbanek, J.; Li, M.; Szlam, A.; Rocktäschel, T.; and Weston, J. 2021. How to Motivate Your Dragon: Teaching Goal-Driven Agents to Speak and Act in Fantasy Worlds. *Proceedings of the 2021 Conference of the Association for Computational Linguistics: Human Language Technologies*, 807–833.

Antimirov, V. 1996. Partial derivatives of regular expressions and finite automaton, constructions. *Theoretical Computer Science*, 155(2): 291–319.

Chittaro, L.; and Serra, M. 2004. Behavioral programming of autonomous characters based on probabilistic automata and personality. *Computer Animation And Virtual Worlds Comp. Anim. Virtual Worlds*, 15: 319–326.

Dill, K. 2011. A Game AI Approach to Autonomous Control of Virtual Characters. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*.

Ekman, P. 1993. Facial expression of emotion. *American Psychologist*, 48: 384–392.

Hanheide, M. 2010. A Framework for Goal Generation and Management. *Proceedings of the AAAI Workshop on Goal-Directed Autonomy*.

Iovino, M.; Scukins, E.; Styrud, J.; Ögren, P.; and Smith, C. 2020. A Survey of Behavior Trees in Robotics and AI. *Elsevier*.

Muñoz-Avila, H. 2018. Adaptive Goal Driven Autonomy. *Case-Based Reasoning Research and Development. ICCBR*, 11156: 3–12.

Neufeld, X.; Mostaghim, S.; and Brand, S. 2018. A Hybrid Approach to Planning and Execution in Dynamic Environments Through Hierarchical Task Networks and Behavior Trees. *Proceedings of the Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 14(1): 201–207.

Nicolau, M.; Perez-Liebana, D.; O'Neill, M.; and Brabazon, A. 2017. Evolutionary Behavior Tree Approaches for Navigating Platform Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(3): 227–238.

Orkin, J. 2004. Symbolic Representation of Game World State: Toward Real-Time Planning in Games. *https://www.aaai.org/Papers/Workshops/2004/WS-04-04/WS04-04-006.pdf*.

Wilkins, D. 1984. Domain-independent planning Representation and plan generation. *Artificial Intelligence*, 22(3): 269–301.

Winter, K.; Hayes, I.; and Colvin, R. 2010. Integrating Requirements: The Behavior Tree Philosophy. *2010 8th IEEE International Conference on Software Engineering and Formal Methods*, 41–50.