

Beyond Nonlinear Animation

Rama Bindiganavale, Jan Allbeck, Liwei Zhao, Jianping Shi
Seung-Joo Lee, Hogeun Shin, Aaron Bloomfield and Norm Badler
Computer and Information Science Department
University of Pennsylvania, PA 19104-6389, USA

In this sketch, we describe a Parameterized Action Representation(*PAR*) software system developed to bridge the gap between natural language instructions and the virtual agents who are to carry them out. Providing a virtual human [?, ?] with human-like reactions and decision-making is more complicated than just controlling its joint motions from captured or synthesized data. Here is where we need to convince the viewer of the character's skill and intelligence in negotiating its environment, interacting with its spatial situation, and engaging other agents. This level of performance requires significant investment in non-linear action models.

One such model is Parallel Transition Networks or PaT-Nets where the network nodes represent processes and arcs contain predicates, conditions, rules, or other functions that cause transitions to other process nodes. The benefits of PaT-Nets arise not only from their parallel organization and execution of low level motor skills, but also from their conditional structure. Traditional animation tools use linear time-lines on which actions are placed and ordered. A PaT-Net provides a non-linear animation model, since movements can be triggered, modified, or stopped by transition to other nodes. This is the first crucial step toward autonomous behavior since conditional execution enables reactivity and decision-making capabilities.

Even with a powerful set of motion generators and PaT-Nets to invoke them, there remains a challenge to provide effective and easily learned user interfaces to control, manipulate

and animate virtual humans. Interactive point and click systems work now, but with a cost in user learning and menu traversal. Such interfaces decouple the human participant's instructions and actions from the avatar through a narrow and ad hoc communication channel of hand motions. A direct programming interface, while powerful, is still an off-line method that moreover requires specialized computer programming understanding and expertise. The option that remains is a natural language-based interface.

The key to linking language and animation lies in constructing Smart Avatars that understand what we tell them to do. This requires a conceptual representation of actions, objects, and agents which is simultaneously suitable for execution (simulation) as well as natural language expression. We call this architectural level the Parameterized Action Representation or PAR [?]. It must drive a simulation (in a context of a given set of objects and agents), and yet support the enormous range of expression, nuance, and manner offered by language [?]. The PAR gives a high level description of an action that is also directly linked to PaT-Nets which execute movements. A PAR is parameterized because an action depends on its participants (agents who execute the action; a list of physical objects involved in the action, and other attributes) for the details of how it is accomplished. A PAR also includes applicability conditions and preparatory specifications that have to be satisfied before the action is actually executed. The action is finished when the terminating conditions are satisfied. Uninstantiated PARs (UPARs) are stored hierarchically in a database (called the Actionary). During execution, an UPAR is instantiated into an IPAR (Instantiated PAR) with specific information on agent, physical object(s), manner, terminating conditions, etc.

The PAR architecture includes the Actionary, a NL2PAR module which converts a natural language instruction to an IPAR and an agent process module. Each agent is controlled

by a separate agent process, which maintains a queue of all IPARs it is to execute. Individual action capabilities and planning abilities may vary across agents. We use the EAI/Transom Jack toolkit and OpenGL to maintain and control the actual geometry, scene graphs, and human behaviors and constraints.

We illustrate the PAR system through a Virtual Environment Training scenario. The purpose of the scenario is to help train military peace-keeping personnel operate a checkpoint. The job of the trainees is to watch for a suspected terrorist. All the virtual agents and the vehicle generating process simulator are defined as individual agent models that act autonomously or in response to messages received from each other or to changes in the environment. Work is in progress to extend this system to work in a distributed environment with heterogeneous clients on different computing platforms.