Pattern of Life

MAK's Approach to Realistic Background Traffic

What is Pattern of Life?

While many of us may look out our car window on the way to work at a crowded city street and see randomness, there is really quite a bit of order beneath the surface. It turns out people – by their very nature – are quite predictable. We take the same car at the same time, on the same road, to work every day. Most people go to work in the morning and return in the evenings; vendors open shops, buses follow schedules, and children go to school all in a pattern. We wear winter coats when it is cold outside, and we wear shorts when it is hot. No one is a robot, and many individuals have some degree of variation in their daily routines, but when taken in aggregate, we pretty much do the same thing at the same times every day. Most of us pay no attention to the pattern in the car on the road to work - it all looks like noise.

These patterns occur around us every day. No matter what you are doing, there is some pattern there: it could be cars passing you on the highway, planes flying above you, people walking next to you. Your world is full of patterns.

Patterns for Surveillance

As the patterns of our daily lives blend together and blur into background noise, sometimes single actions start to stand out. It seems that, eventually, we get used to the noise. In the world of Intelligence, Surveillance, and Reconnaissance (ISR), intelligence agents (people and now machines) are trained to spot deviations from the standard pattern: they find the signal in the noise and act on it.

This is not a new concept. Try driving around in the rain with your drivers-side car window down. A police officer will likely stop you assuming the car is stolen and the window was smashed. Driving with open windows in the rain is not part of the standard driving pattern. Wear a winter coat into an airport when the temperature is 100o F degrees and you will likely be approached by security quickly.

Modeling Pattern of Life is Important

Even if your end goal is not related purely to surveillance, almost any scenario you can imagine should include background traffic. If you want to model a car on a highway, you must consider the other vehicles, animals, and pedestrians that may be on the highway at the same time. If your focus is the highway and your goal is to build an immersive driver trainer, you should consider adding the occasional plane in the sky. If you're interested in port security, you may need to consider adding not just boats to the scene, but cars and trucks traveling over the bridge at the harbor entrance. What about the subway station exit which exists right next to the Olympic venue? How many passengers will be exiting the station, and at what frequency? Without generating the expected background noise, your scenario will have less value and appear unrealistic to everyone.

Modeling Pattern of Life is Hard

Pattern of Life modeling is hard for two reasons: volume and complexity.

Pattern of life requires lots of simulated entities. While some patterns can occur when there are very few people around (the people working in the abandoned village), most occur in busy villages or bustling city streets. Simulating enough simultaneous entities to create the noise of a busy airport is complicated. Most simulators cannot keep up with maintaining the state of all the actors needed to compose a full scenario, even if you only need to see a small portion of them at any one time.

Within the overall pattern are many individual behaviors. Some behaviors are quite complex. Typically, the behavior of our daily activities is complex enough that when coupled with large numbers of entities, the cost becomes prohibitive. It is not hard to model one individual shopping on a New York City street, but what about one thousand people shopping on a street with hot dog vendors, police walking around, and tourists taking pictures? Setting up such a scene could take weeks of scripting. Even simpler tasks like modeling a busy port (there are many fewer boats entering the Port of New Orleans than there are people on 5th Avenue at noon on a hot sunny day) take quite a bit of effort. Making a string of tankers enter the port is one thing, but giving them some degree of randomness is harder.

Unfortunately, this background traffic is often the first thing that gets cut when time gets tight. Yet, the presence of this background traffic is the key to selling your demo, and the difference between a good scenario and a weak one.

VR-Forces Supports Pattern of Life Modeling

VR-Forces, MAK's computer generated forces (CGF) application and toolkit provides a robust pattern of life capability. You can quickly create individual entities that can move through the scenario using default behaviors or custom plans.

While creating individual entities is useful, especially when modeling the target, real pattern of life modeling requires streams of semi-randomized people and objects following a pattern. Without randomization, the pattern becomes too obvious. We do not actually end up at the same intersection with the same people at the same time every day we go to work. VR-Forces lets you create the pattern through an intuitive user interface and then instantiate the pattern without the requirement to carefully place each individual person who exists in the pattern.

Creating these patterns is straightforward. You create instances of Spawn points in your scenario and then combine multiple spawn points to generate complex streams. The Spawn point specifies one or more entity types that are created and then execute either generic movement or a custom plan, which dictates their behaviors and eventual removal from the scene.

Entity Recycling

With most patterns, the key is diversity and large numbers of people or vehicles. Early pattern modeling involved a fixed set of objects, which moved around repeatedly in a scene. Creating new objects only occurred at the beginning of the scenario and each object lived throughout the whole life-time of the scenario. While this can meet some pattern of life model needs, it is often impossible to simulate the volume (perhaps thousands) of objects needed to produce the "randomness" required for an actual specific scenario. With VR-Forces' spawn and sink process, entities enter the scenario, move through an area of interest and then are removed from the scenario. This creates a continuous natural stream of entities through the area of interest, without the need to simulate the total number of entities for the entire length of the scenario.

VR-Forces lets you focus on the creation of the pattern through intuitive user interfaces, without worrying about managing the behavior of any particular entity at any particular time. For example, suppose you want to model traffic at a subway station entrance. Trains start running at 05:00, but traffic builds at rush hours from 07:00 to 10:00 and 16:00 to 18:00.

Trains arrive approximately every 10 minutes, with bursts of traffic leaving the station after the train arrives. Trains are not always on time, and people group differently leaving the station. For most of this time, people are entering the station at a somewhat continuous pace of about 6 to 10 people per minute.

Figure 1 Subway traffic

To do this, you need to model people entering the station and leaving the station. They need to do this at different volumes depending on the time of day. You can do this by designing about 4 patterns, yet over the course of the day many thousands of entities may appear in the scene.

Spawn Patterns

In VR-Forces, you configure patterns of life as spawn patterns. You can have multiple spawn patterns in a scenario. Each spawn pattern specifies the following pattern variables:

- Entity type - the types of entities to create and the maximum number to simulate at one time.
- Spawn timing - continuous, by time of day, by simulation time.
- Periodic bursts - spawn entities for some percentage of a given time interval.
- Spawn interval - the amount of time between creation of each entity.
- Spawn interval variance - add variability to the spawn interval.
- Default plan - entities move to a randomly chosen sink point.
- Custom plan - entities follow a custom plan that can include any of the tasks, sets, and conditions available to plans in a scenario.

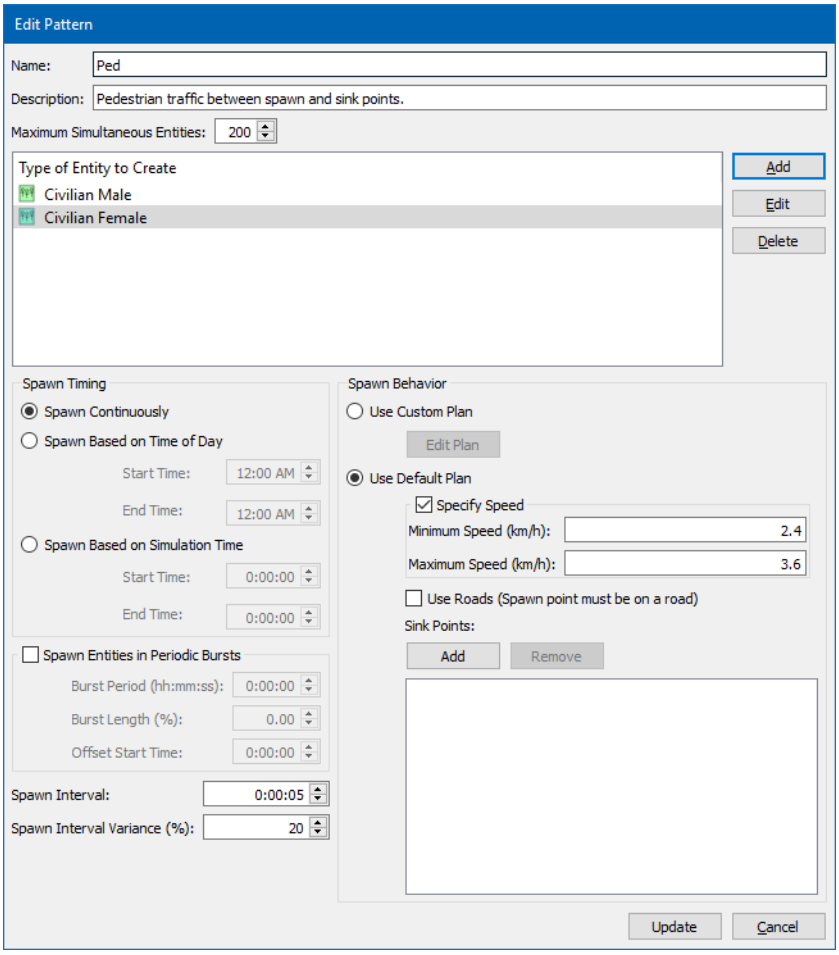

Figure 2 Spawn Pattern Manager

Example Patterns

Ground Vehicle Traffic

- Randomized traffic following standard routes, which is time sensitive. Traffic builds as rush hour commences.
- Follow specific roads, but also have slightly randomized paths. That is, we can say 20% of traffic will take a right at intersection A, and 80% goes straight, with 50% turning right at intersection B.
- Shaped vehicle traffic on select roads (Intersection of A and B street). Most people turn right from A>B in the morning and left from B>A in the evening.

Public Transportation

Public Transportation arrives at a station every 10 minutes from 08:00 to 10:00 and 16:00 to 18:00. From 10:00 to 16:00, the transportation arrives at 20 minute intervals. People leave the station with a burst after every train arrives, the bursts are somewhat random.

Animal Traffic

- A donkey or camel walks through your scene at various random intervals (every couple of hours with a .23 variance).
- A herd of sheep crosses from the pasture on the east of the road to the pasture on the right of the road in the morning around 08:00 and crosses back at night.

Sophisticated Human Traffic

- Sidewalk traffic that is weather and time of day specific – shoppers at midday and commuters in the morning and evening.
- School children at 09:00 entering the school, and leaving at 15:00.
- Complex behaviors like lunch breaks: Entities leave building A and leave the scene through a semi-random path.

Air Traffic

- Planes land and take off in a semi-randomized pattern. Planes land, find a gate, wait some time between 20 and 30 minutes, and then take off again.
- Planes cross the sky in specific patterns – appearing approximately once an hour.

Maritime Traffic

- Cargo ships enter harbors occasionally – they tend to enter in the morning, and leave in the evening.
- Pleasure craft “wander” around the harbor.
- Realistic background traffic flowing through the Red Sea or Persian Gulf. Entering traffic builds in the morning and slows down in the afternoon. No ships transit at night.

Using Crowds for Background Traffic

In addition to using its pattern of life feature to create background traffic, VR-Force supports dedicated civilian crowds, whose members can wander through an urban terrain, paying attention to sidewalks, crosswalks, and buildings. You can create crowds as units of varying sizes, without the need to place members individually.

Members of a crowd can react automatically to certain events, such as an explosion. You can task them to gather around an entity or location of interest. They can protest or flee from an entity.

The crowd behaviors are controlled by scripted tasks, so you can change their behaviors or add to them using VR-Forces' Lua scripting capability.