

# An Introduction to Simio<sup>®</sup> for Arena Users

---

C. Dennis Pegden, Ph.D.

This white paper is intended to help experienced Arena users transition from Arena to the new Simio simulation product. As an experienced Arena user you have many advantages over a new user in terms of modeling experience and project management. However care is required to ensure that your knowledge of Arena does not become a handicap that prevents exploiting the full power of Simio. Although Simio is very simple and powerful – that power can best be leveraged by a different approach than you are accustomed to with Arena. To evaluate Simio, link to [free-software-registration](#).

## The Arena Perspectives

Arena models are built using a single modeling paradigm referred to as a **process orientation**. You define **elements** that hold the state of the system, and build process flows using **blocks** that perform actions on the elements. The blocks are passive and are activated by the arrival of an **entity**. Entities move from block to block and change the state of the model over time. The elements, blocks, and entities define the process model.

You animate your model in 2D as a 2-step process. First you draw process flows for your model, and then in a separate area of the same drawing space you draw the animation for the facility. This is sometimes done by first importing an AutoCAD drawing as a static background for your animation. You then add levels, animated routes, etc. that are linked back to your process model. The tasks of modeling and animating your model are typically done separately.

You can employ Arena template panels to place hierarchical modules in your process flow. These modules are used in the same way as the standard blocks; however they represent a user-defined sequence of blocks. For example you can create a Process module that is comprised of Seize-Delay-Release. The Process module can have properties (e.g. processing time) that are passed to the Seize-Delay-Release blocks.

## The Simio Perspective

Although it is possible to build process models in Simio, models are typically built using an object orientation. You drag objects into your **facility view** and connect them together in a 3D drawing space. The objects define both the logic for the model, and the animation. Modeling and animation are done as a single step.

Modeling with objects requires a shift in perspective. In the Arena process view you think in terms of a logical process comprised of blocks that are activated by an entity. The blocks represent logical actions

such as seizing a resource, delaying by time, etc. In the Simio object view you think in terms of the physical objects in the system – i.e. the machines, robots, conveyors, etc. that make up the system. You place these objects in your facility model and they interact based on their internal logic.

In Arena when you refer to your “model” you are referring to the Arena process model. This is the only modeling approach supported in Arena. However Simio is a multi-paradigm modeling tool that supports both an object orientation and process orientation. You build an object-based model in the **facility window**, and/or a process based model in the **process window**. The facility window is a 3-D drawing space in which you place objects that represent the physical components of your system. The process window is a 2-D drawing window in which we construct Arena-like process flows. Although the object paradigm is the primary modeling approach in Simio, you can freely mix the two modeling approaches together in the same overall model. We will sometimes refer to these two separate components of the overall model as the **facility model** and the **process model**.

The objects that you place in the facility model are user-defined and are stored in libraries. As an Arena user you may be tempted to think of these object libraries as similar to Arena template panels; however they are quite different. An Arena template panel is a library of hierarchical blocks (modules) that can be placed in process logic. In contrast a Simio library is a collection of object definitions for placing objects in your facility model. In other words a Simio library contains object definitions that represent the physical components of a system, and an Arena template panel contains logical constructs for use in a defining a logical process flow. The Simio libraries and Arena template panels share the basic notion of user-defined library, but they differ in the modeling orientation that they are designed to support.

There are five different object **classes** in Simio and each has a defined role within the Simio modeling framework. A **Fixed** object has a defined location in the model and is used to represent things such as machines, work stations, assembly areas, etc. A **Node** object defines the starting and/or ending location for one or more Link objects. A Node object also supports the transfer into and out of an associated Fixed object. Node objects are used to represent traffic intersections, conveyor merge points, etc., as well as entry and exit points for Fixed objects. A **Link** object moves an entity over a pathway connecting two Node objects. Link objects are used to represent things such as conveyors, AGV tracks, roads, etc. **Entity** objects can be dynamically created and destroyed, move across a network of links and nodes, move through 3D space, and move into and out of Fixed objects. Examples of Entity objects include customers and work pieces. **Transporter** objects are a special type of entity that can also pickup, carry, and drop-off other entities. Examples of Transporter objects include carts, trucks, and buses.

It is important to note that the classes in Simio define a generic behavior, but not the specific behavior of an object. The specific behavior of an object is defined by an **object definition** which gives the object its own intelligent behavior. For example we have a Link class for defining a pathway between two nodes, and we can have a Conveyor object definition that is created from the Link class, but has specific logic added to represent the behavior of a conveyor. An object definition begins with some generic behavior defined by its class, and then refines that by modeling the specific behavior desired for that

object. Object definitions can also have properties that further customize the behavior. Once you have created an object definition you can then place **objects** with the defined behavior into your facility model. An object is a specific instance of an object definition that has been placed within a model.

Simio comes with a Standard Object Library that includes 14 pre-built object definitions that can be used to model a wide range of systems. This library is briefly summarized in the following table.

Name	Class	Description
Source	Fixed	Creates entities that arrive to the system.
Sink	Fixed	Destroys entities and records statistics.
Server	Fixed	Models a multi-channel service process with input/output queues.
Resource	Fixed	Models a resource that can be used by other objects.
Combiner	Fixed	Combines entities in batches.
Separator	Fixed	Separates entities from batches.
Workstation	Fixed	Models a 3-phase workstation with setup, processing, and teardown.
Vehicle	Transporter	Carries entities between fixed objects.
BasicNode	Node	A simple intersection of links.
TransferNode	Node	An intersection where entities set destination and wait on transporters.
Connector	Link	A zero-time connection between two nodes.
Path	Link	A pathway between two nodes where entities travel based on speed.
TimePath	Link	A pathway with a specified travel time.
Conveyor	Link	An accumulating/non-accumulating conveyor device.

As we will discuss in more detail shortly, each of these object definitions are really just models that define the behavior for the objects. You build object definitions the same way you build any model in Simio. In fact any time you build a model of a system, that model can serve as an object definition. You can then place objects with the behavior defined by that model into other facility models.

Note that the library contains a Link object definition named Conveyor that can represent both accumulating and non-accumulating devices. In Arena a conveyor is a specific construct that is built into

the Arena engine and cannot be changed by you as a user. In contrast the Conveyor in this library is an object definition and you can change and modify its behavior as needed by changing its underlying model. The specific conveyor defined in the Standard Object Library supports both fixed and variable spacing and accurately models the merging of entities of various sizes on the conveyor.

Note also that the library contains a Transporter object definition named Vehicle. This particular object definition models a transporter device that can carry multiple items between pickup and drop-off locations. You can create other transporter models that behave differently than the standard Vehicle model; e.g. you could create an object definition for a subway car, AGV, elevator, or bus.

As a Simio user you can create your own object definitions from any of these five classes. For example you might build a set of object definitions for modeling emergency rooms that include a MRI (Fixed), Triage Station (Fixed), Doctors (Entity), Beds (Transporter), Hallways (Link), etc. As we will discuss shortly you can create models for these object definitions from scratch, or you can modify or extend the models for existing object definitions such as those provided by the Standard Object Library.

## Simio Entities

The “things” that move through the system in Simio are referred to as entities. However the entities in Simio are very different from the entities in Arena. In Arena entities are part of a process model of the system and are completely dumb; their only purpose is to carry information (attributes) and to execute a process. In Simio entities are part of an object model and can have their own intelligent behavior. Entities can make decisions, reject requests, decide to take a rest, etc. Entities have object definitions just like the other objects in the model.

Entities in Arena can have attributes; in fact every entity in an Arena model must have exactly the same attributes. In Simio you can have many different types of entities, and each one may have its own set of Properties and States which carry data and help define its behavior. Unlike Arena, that same principle applies to other constructs as well - every object may have its own set of Properties and States which carry data and help define its behavior.

In a moment we will discuss process modeling in Simio. At that time we will introduce the concept of a “dumb” token that carries information and executes a process. An Arena entity corresponds to a token in Simio.

## Simio Resources

Both Arena and Simio have the concept of resources. However in Simio the concept is quite different and much more robust. Any object in Simio can seize and release any other object. For example a Server object could seize and release a Worker object. However a seize operation requires a negotiation between the objects involved. In the above example the Worker object could refuse the seize request

because he/she has other activities to perform. Hence in Simio objects allow or disallow themselves to be seized and released by other objects.

## The Role of Processes in Simio

As an Arena user you have built your models using a process orientation. In Simio the process orientation still exists, however its role within model building is completely changed. In Simio the process orientation is used to either customize the behavior of an existing object, or to create new object definitions.

A process in Simio is very similar to a process in Arena, with slightly different terminology. A process in Simio is comprised of **steps**, **elements**, and **tokens**. Tokens flow through a process executing steps that alter the state of one or more elements. Hence steps are like Arena blocks, and tokens are like Arena entities.

You can build many simple models in Simio by relying totally upon the object definitions from the Standard Library. However once you start modeling complex real-world examples, you will likely encounter the need to use your Arena-like process modeling skills to extend the behavior of these objects. This is one of the primary roles of process modeling in Simio.

When you place an object from the Standard Library into your facility model, you can have the object run “add-on” processes at specific points in the logic. An “add-on” process is simply a logical process that you have created to perform some specific action such as assigning a state variable, seizing a resource, referencing information in a table, etc. For example the Source object has four “add-on” processes: Initialized, Creating Entities, Created Entity, and Exited. You can add your own custom logic to this object at any of these four points in the object logic.

The second major role for process modeling in Simio is for creating new object definitions. An object definition is simply a model for how instances of that object should behave. These models can be built in three primary ways. The first is to build facility model using one or more existing object definitions; i.e. traditional hierarchical modeling. For example you might combine two servers in series into a new object definition representing tandem servers. The second way is to define the object using a set of processes. Objects built in this way are referred to as base objects. This is the most flexible and powerful approach and is the method used for creating the object definitions in the Standard Library. The third way is referred to as sub-classing. In this case you inherit the behavior of an existing object and then modify it by overriding or adding one or more processes.

Once you successfully make the transition from a process modeling perspective to an object modeling perspective your Arena-like process modeling skills will be very useful for both enhancing the use of the Standard Library objects, and also creating new object definition libraries for use in your modeling activities.

## Summary

As an Arena user switching to Simio you will be faced with the challenge of changing your modeling perspective to an object-based approach. Model your system from the perspective of the “things” that make up the system, and not the logical process flows within your system. Employ your process modeling skills for the purpose of customizing the logic for an existing object (i.e. add-on processes), or for building libraries of new object definitions for use in your models. Once you have mastered this new approach you will be able to fully exploit the power of Simio.

### About The Author

Simio® LLC was founded by Dennis Pegden, Ph.D. to reinvigorate the field of simulation modeling. A 33-year veteran of the simulation industry, Pegden’s career highlights include leading the development of SLAM (marketed by Pritsker and Associates) and founding Systems Modeling Corporation, now part of Rockwell Automation. Pegden also led the creation of the simulation products SIMAN and Arena. Simio’s management team brings one hundred five years of combined experience in the simulation market. Contact Dennis Pegden at [dennis\\_pegden@simio.com](mailto:dennis_pegden@simio.com). To evaluate Simio, link to [free-software-registration](#).