



JSIMMOD - A New/Old Beginning

NASUG Conference, September 21, 2001

Presented by:
Gregory Bradford
AirportTools

Goal

An aviation simulation platform that

Adheres to established simulation criteria, and

Provides a foundation for future development

While promoting

Ease of usage,

Extensible model development,

Affordable acquisition and application costs,

Stable and repeatable results,

A maintainable and open source code base, and

Cross platform compatibility.

Why JSIMMOD?

✈ A need exists for a high performance, feature rich, and cost effective aviation simulation model.

✈ None of the existing models fulfill all of the above needs.

Either:

✈ It costs too much.

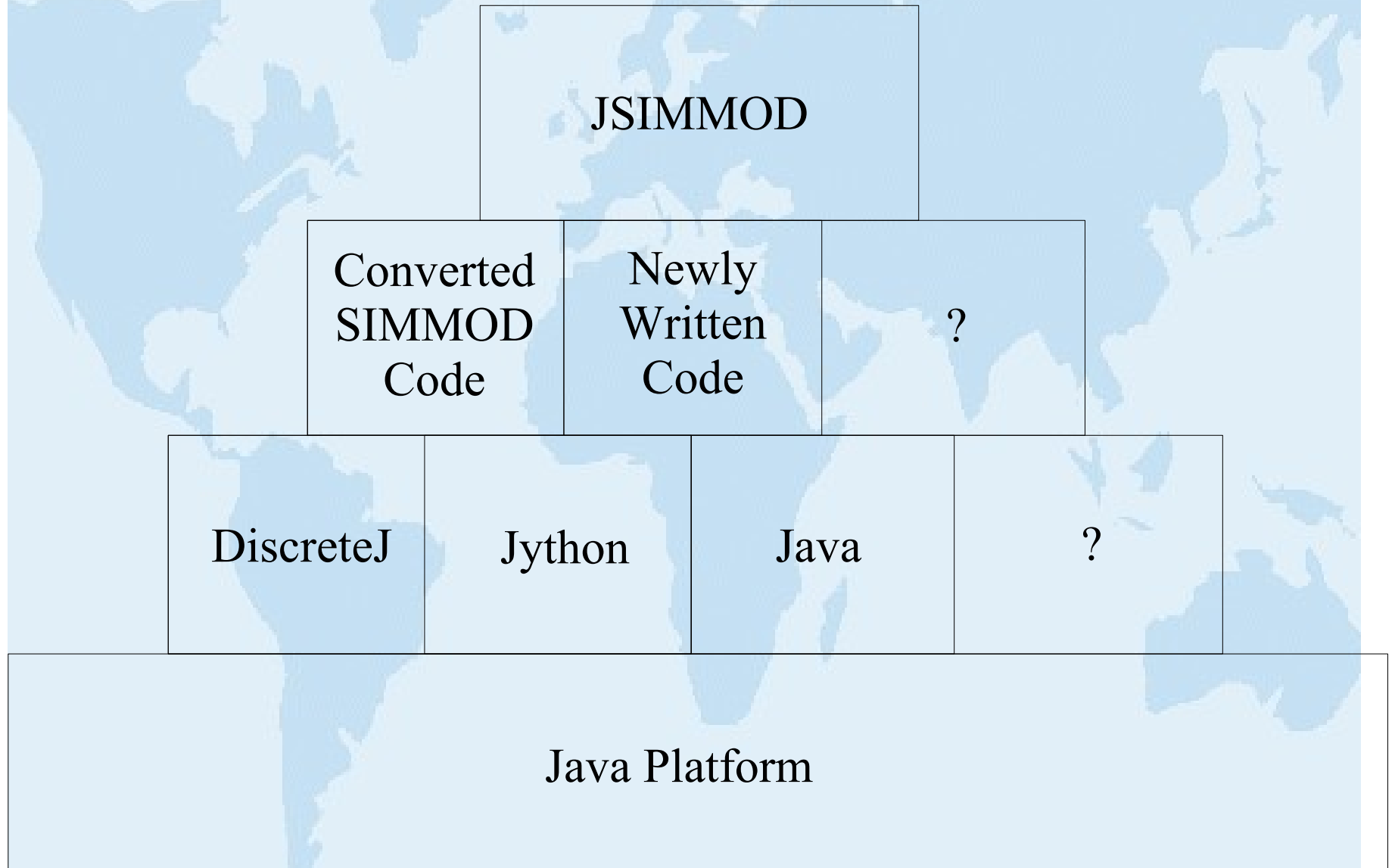
✈ It doesn't perform everything you need to do.

✈ You cannot have it.

What is JSIMMOD?

- ✈ JSIMMOD is an in progress re-implementation of SIMMOD. Thus, retaining SIMMOD's functionalities and maintaining investments in SIMMOD expertise, applications, software and name recognition.
- ✈ JSIMMOD is a new program with capabilities that will rival and surpass the capabilities of a majority of the other comprehensive aviation simulation models.
- ✈ JSIMMOD is a program that will be made widely available at no cost, with it's source code, so that, if it doesn't do what you need it to do, you will have the option of enhancing or fixing it.

What is JSIMMOD (continued)



JSIMMOD Costs

✈ Software Costs:

- ✈ Java – Free, i.e. no acquisition costs, no distribution costs.
- ✈ Jython – Free, i.e. no acquisition costs, no distribution costs.
- ✈ DiscreteJ – Free, i.e. no acquisition costs, no distribution costs.

✈ Development Costs:

- ✈ AirportTools will contribute to the development of JSIMMOD for free.
- ✈ SIMMOD serves as a template for parts of JSIMMOD, thus reducing development costs (or rather development time).

✈ Support Costs:

- ✈ AirportTools will support JSIMMOD on an as-available basis, for free.

A Fine Match

- ✈ When combined, Java, Jython, and DiscreteJ form a very powerful environment (JSIMMOD) for modeling aviation operations.
- ✈ Java is a very modern object oriented computer language which has become the most popular programming language in just five years.
 - ✈ Java is widely supported by third parties.
 - ✈ Java is network ready. It can easily "talk" to other programs.
 - ✈ Java's strict syntax promotes good programming practices.
- ✈ Jython is a Java implementation of the popular scripting language Python.
 - ✈ Jython is widely used, by analysts and programmers alike.
 - ✈ Jython is open source, or rather it is free.
 - ✈ Substantial Python documentation exists.

A Fine Match (continued)

- ✈ DiscreteJ is a "Best of Breed" process oriented discrete event simulation library designed to replace Simscript II.5.
 - ✈ Eight different discrete event simulation libraries were thoroughly reviewed and tested to gather appropriate features.
 - ✈ FIFO, LIFO, and Priority Based Queues are supported in DiscreteJ. Additional queue types easily added if needed.
 - ✈ Aircraft can simultaneously belong to multiple queues and queue types in DiscreteJ.
 - ✈ Queues and distributions have built in recording and reporting capabilities. Spreadsheet formatted logs are created on demand.
 - ✈ Integer linear, real linear, and real exponential probability distributions are currently supported. Other distributions can very easily be added as needed and/or requested.

So What Does All This "This Oriented" and "That Oriented" Really Mean?

- ✈ Greater modeling flexibility. For example, at a various run-times, you can direct JSIMMOD to run your Jython scripts that might:
 - ✈ Change the number of aircraft allowed to park at a gate.
 - ✈ Add/Remove (or turn on/off) a set of arrival staging areas.
 - ✈ Apply a specific taxipath determination logic.
 - ✈ Detect a gridlock situation and direct A/C to taxi differently.
 - ✈ Build an entirely new route in the model, based on some criteria.
 - ✈ Create your own pilots and controllers. Have their communications, or lack thereof, impact the model.
 - ✈ E-mail you that the model has reached a gridlock situation.
 - ✈ Write the status of some model aspect to your "special" log file.

So What ... Really Mean (continued)

✈ Larger Models.

✈ No arbitrary limits on the number of links, nodes, or routes. For example: Model the entire United States.

✈ Link models, using Jython and Java's networking capabilities. Have one instance of the model simulating Western Europe, another Atlantic air routes, and another the Eastern United States. Have all of them communicating and interacting.

✈ Greater useability.

✈ JSIMMOD's object-oriented design ensures fewer bugs from the start.

✈ Can easily use reasonable filenames, rather than SIMU01, ...

✈ Automated documentation generators (built into Java) ensure substantially more documentation for users and developers.

So What ... Really Mean (continued)

- ✈ Work smarter and faster.
 - ✈ JSIMMOD can check/verify much of the data that is done now by third-party tools.
 - ✈ Edit Jython scripts as the model is running.
 - ✈ No compilation necessary for Jython.
- ✈ More choice.
 - ✈ A wider variety of organizations will enhance JSIMMOD.
 - ✈ Java third-party support is substantial, Simscript II.5 third-party support is non-existent.
 - ✈ Run JSIMMOD on a Mac, Linux, or PC box. Even run JSIMMOD from a web page as an applet.

Current Status

- ✈ DiscreteJ is working well. Since DiscreteJ is necessary for JSIMMOD, AirportTools will make it available for no charge.
- ✈ Jython has been tested to confirm it works in the desired manner.
- ✈ No discrepancies noted between Java's floating point number arithmetic operations and Simscript II.5's. This is critical for duplication of results.
- ✈ JSIMMOD's execution times are comparable to SIMMOD's.
- ✈ Tested approximately 5,000 aircraft simultaneously aloft on a 256mb computer to confirm ability to handle large models.
- ✈ SIMU01, SIMU03, and SIMU07 flat file reading routines finished.
 - ✈ Can be used as-is as a secondary input data checking program.

Current Status (continued)

- ✈ Extensive and robust SIMU02 output available now.
 - ✈ PRINT 1's for all cards, SIMMOD has only partial support. JSIMMOD's PRINT 1 format and content closely matches SIMMOD's, but errors were corrected when found.
 - ✈ PRINT 2's and 3's implemented for extended debugging support.
 - ✈ Standardized error/warning messages (HTML?). Easy searches.
- ✈ Basic airspace working.
 - ✈ Aircraft traverse their assigned route and the corresponding link and node structure properly.
 - ✈ Accounting for Level I delays seems to be correct in most instances. Work continues on this aspect.

Schedule

- ✈ Finish the basic airspace logic (2 man months).
- ✈ Basic runway/procedures logic (2-4 man months).
- ✈ Basic taxiing, staging, and gating logic (4 man months).
- ✈ Incorporate user defined Jython scripts into as many areas as can logically be provided (time will vary).
- ✈ Build broadcaster and listener logic so that human and communications elements can be studied in unison with a valid airspace and airfield model (3 man months).
- ✈ Add a basic free-flight capability to JSIMMOD (2-4 man months).
- ✈ Add more esoteric aspects of SIMMOD's aviation modeling, such as runway u-turns (? time).