NPL Specimens Simulation

The NPL Specimens simulation models the photography and imaging of fossil specimens. The modeling software is *SimPy*, a Python library, with supporting graphics routines from the Python *MatPlotLib* library.

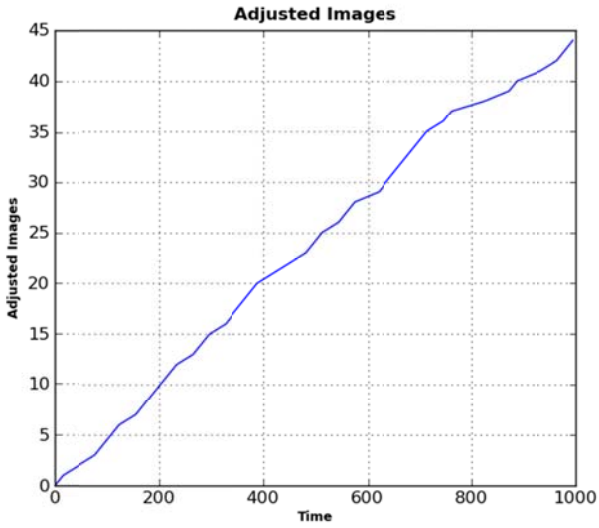
The *specimens.py* model has three main components: (1) processes, (2) resources, and (3) stores. Processes represent actual steps in imaging specimens: photograph, conversion, Helicon focus, and final PhotoShop processing (brightness/contrast adjustment and conversion to final image formats). Resources are cameras, computers, and people. Stores represent bins or directories representing sets of specimens or images in various stages of processing.

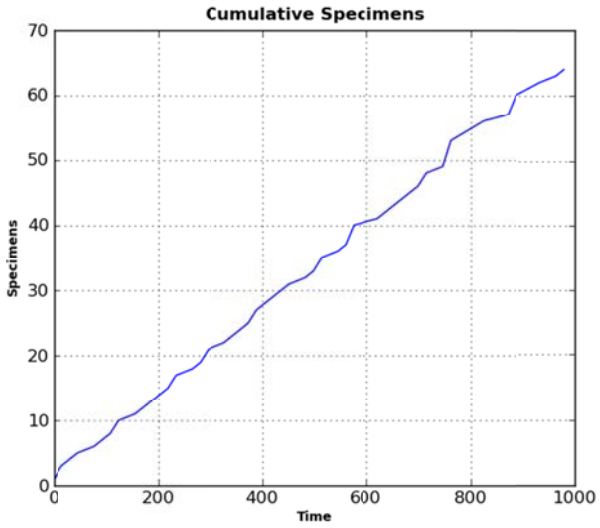
A process requires particular resources and a source of input (from one of the stores), and additional requirements (such as the states of all the stores) may be specified.

A single simulation is based on sets of specimens to be imaged. The stochastic nature of the imaging project is modeled by drawing the number sets from a uniform distribution (between 11 and 50 catalog numbers), the number of specimens per set from an exponential distribution with the same mean as found in the actual collections, and processing times from Pareto distributions with minimum times representing "best case" estimates.

Output from an example run is shown below

```
0 min - Photographing
4 min - Converting
9 min - Heliconing
12 min - PhotoShoping
15 min - Photographing
24 min - Converting
34 min - Heliconing
41 min - PhotoShoping
        .
        .
        .
        .
963 min - Photographing
968 min - Converting
973 min - Heliconing
976 min - PhotoShoping
979 min - Photographing
984 min - Converting
988 min - Heliconing
992 min – PhotoShoping
```





In this example there were 44 sets of specimens, with a total of 64 individual specimens distributed by catalog number. The graphs above show the accumulation of processed images in the final set (called Adjusted Images).

The generalized flow chart of a model simulation is shown in the attached figure. Python source code is also attached.

Labels

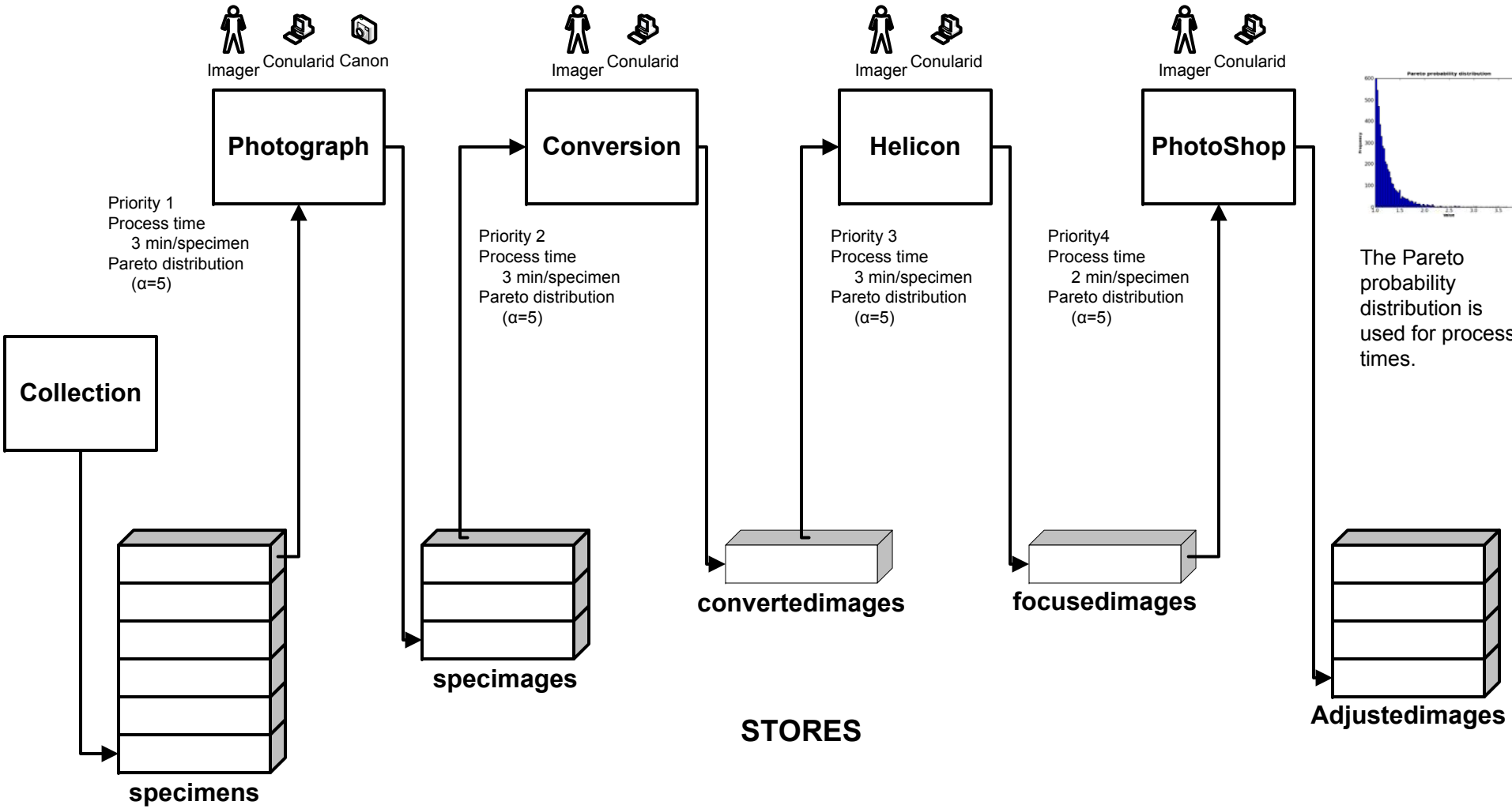
Label scanning is modeled in a similar fashion. The generalized flow chart of label scanning simulation is shown in the attached figure. Python source code is also attached.

System Requirements

The model requires Python Version 2.6 and the additional add-on libraries, SimPy Version 2.0.1, MatPlotLib Version 0.99.1, and NumPy Version 1.4.1. All of these packages are open source and can be obtained via the SourceForge.net website. Executable installation files for the add-on packages have been copied to the npl disk at DocLib\processModels\PythonAddons and DocLib\processModels\SimPySource. Details for installation and configuration are included with the executables.

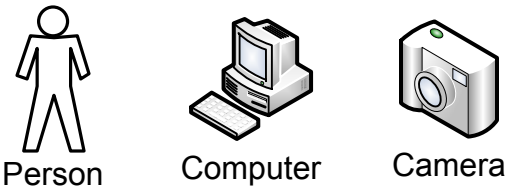
Specimen Photography

PROCESSES



The Pareto probability distribution is used for process times.

STORES

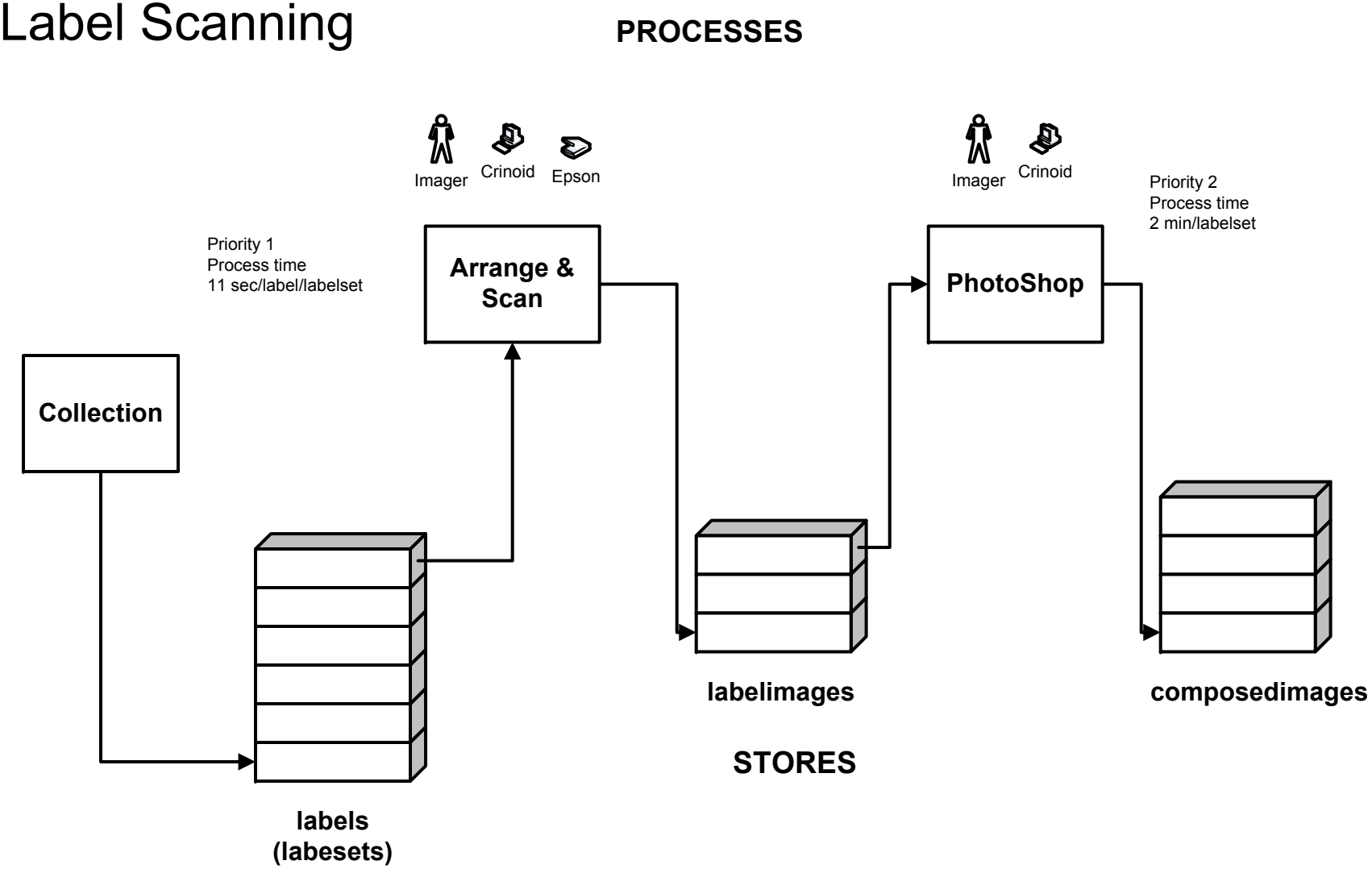


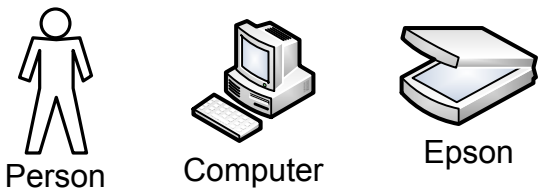
RESOURCES

Processes require available resources and available stores. The *PhotoShop* process has the highest priority for resources, *Photograph* the lowest. In addition, the *Photograph* process proceeds only when the *specimages*, *convertedimages*, and *focusedimages* stores are empty.

The simulation ends when the *specimens* store is empty

Label Scanning





Processes require available resources and available stores. The *PhotoShop* process has the highest priority for resources. In addition, the *Arrange & Scan* process proceeds only when the *labelimages* store is empty.

The simulation ends when the *labels* store is empty

Combined Processes