“Implementing MVC Architecture in Python for Data Analysis”

Ernest Bonat, Ph.D.

Senior Software Engineer

What do I do?

1. Full-time position at Intel Corporation as Senior Software Engineer at USA IT Group (Hillsboro, OR) – working with Microsoft .NET, APEX Salesforce, Python, etc. (8 to 5 pm)

2. Consulting IT Application Development and Data Analysis for clients in needs. (2-3 hours at night)

3. Teach Computer Programming and Business Statistics classes online for fun. (2-3 hours at night)

Main question for me: How to reduce my Python Data Stack libraries installation/update and program development time?

What do I need?

1. A Python IDE for everything (database, file, development code/debugging, source control, etc.)

2. A fast Data Stack libraries installation/update with a possible EXE installation package.

3. A program design pattern (architecture) for fast development and future maintenance.

Example: Need to install pandas! (http://pandas.pydata.org/)

pandas can be installed via pip from PyPI:

```
pip install pandas
```

How about SciPy, NumPy, Matplotlib, etc.? -> **can’t do that, to much typing!**

Good and easy EXE installation package is needed!

Anaconda, a cross-platform (Linux, Mac OS X, Windows) Python distribution for data analytics and scientific computing.

(https://www.continuum.io/downloads) – Updates on Fridays!

Need a **unique** and **free** Python IDE

(https://wiki.python.org/moin/IntegratedDevelopmentEnvironments)

1. **Eclipse IDE / PyDev plugin / EGit plugin (free)**
2. PyCharm (free and commercial)
3. LiClipse (commercial)
4. Python Tools for Visual Studio .NET (free)
5. NetBeans (free)
6. Komodo (commercial)
7. Spyder (free)
8. etc.

Why Eclipse IDE / PyDev plugin / EGit plugin?

1. Interactive Debugging
2. Remote Debugging
3. Code Autocomplete
4. Code Refactoring
5. Code Analysis
6. Unit-test Integration
7. Django Integration
8. Source Control
9. Much more…

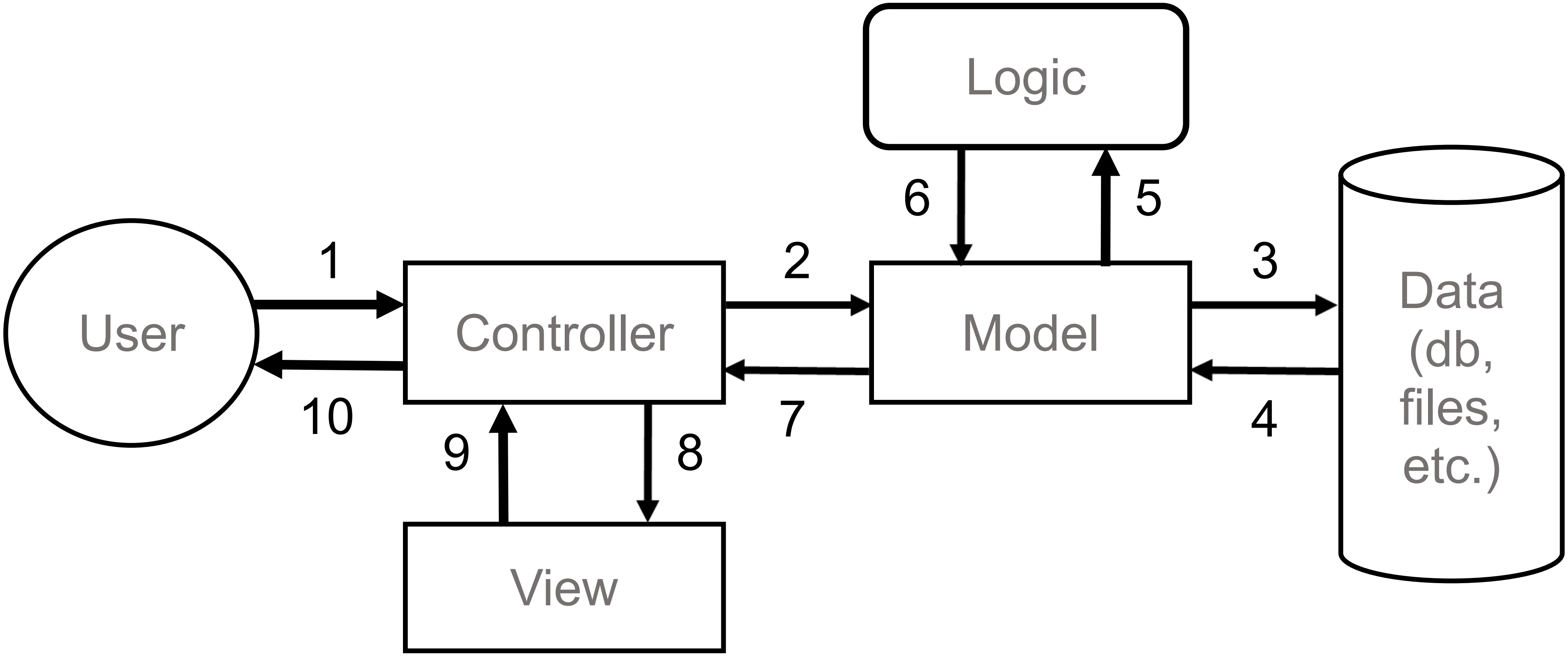
I got my hardware and IDE setup and running:

1. Fast Hardware (Intel Core i7, 16 GB RAM, 64-bit Windows, etc.)
2. Eclipse IDE
3. PyDev plugin
4. EGit plugin
5. Anaconda Installation Package

How to start developing a Python program for Data Analysis? Or may be for anything?

I need to use some simple, standard and fast design pattern!

MVC Architecture (Model-View-Controller)



Who created the MVC architecture?

Trygve Reenskaug (Norwegian computer scientist and professor emeritus of the University of Oslo) – formulated the model–view–controller (MVC) pattern for graphical user interface (GUI) software design in **1979** while visiting the Xerox Palo Alto Research Center.

“MVC was designed as a general solution to the problem of users controlling a large and complex data set”

- Good statement for using MVC for Data Analysis!

Model – Data access and management (CRUD – create, read, update and delete operations) **and provides the business logic operations.**

View – A visual presentation of data (GUI – Graphical User Interface)

Controller – Controls the interactions between the Model and View. It responses to the user input and perform interactions on the data model.

Logic – provides the business logic operations. (**New layer!**)

Project Folders Structure

```
project_name
    src
    package_name
        folder_name
          module_name
```

Example:

```
company_server
    src
    logserver
          controller
            logservercontroller.py (log_server_controller.py)
```

Helper Folders

1. configuration – file.cfg
2. csv – any csv data files (or any)
3. library – generic public functions
4. log – log files
5. test – unit-test implementation
6. project documentation (developer documentation, end-user manual, business requirements, etc.)

Data Analysis is a process of **inspecting**, **cleaning**, **transforming**, and **modeling** data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making.

The Data Analysis process contains, in general, the following ten main logical steps:

1. Business Situation
2. Define Influenced Variables
3. Data Collection
4. Data Processing
5. Data Cleaning
6. Data Presentation
7. Data Analysis
8. Data Transformation
9. Data Conclusion
10. **Making Business Decisions – main result!**

Used Python Data Stack Packages

1. **NumPy** – fundamental package for scientific computing (Numerical Python).

2. **pandas** – provides easy-to-use and high-performance data structures.

3. **matplotlib** – a 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Data Analysis Example: Analyze Company Server Log File!

Time	Priority	Category	Message
10:47.2	Info	Firewall Event	SonicWALL initializing
10:55.2	Error	Firewall Event	Interface X0 Link Is Down
10:55.2	Warning	Firewall Event	Interface X1 Link Is Up
10:55.2	Error	Firewall Event	Interface X2 Link Is Down
10:55.2	Alert	Firewall Event	Interface X3 Link Is Down

Categorical Variables:

1. Server Priority (Message Types): **Info, Error, Warning, Alert**
2. Server Messages: **Information Messages**

Statistical Method: Percent Frequency Distribution – number of observations falling in the percentage of observations.