**ALEXANDRU COLESNICOV**[1]
**LUDMILA MALAHOV**[2]

# USING SECOND GENERATION WIKI-PLATFORM IN THE DEVELOPMENT OF ONLINE ECONOMIC APPLICATIONS[3]

***Abstract.*** *Concept of Web 2.0, also "Participatory Web" or "Social Web", is known since 2004. The main characteristics of Web 2.0 are wide participation of users in creation of the site content, and great interoperability, i.e., compatibility with other systems. The work discusses the essential aspects of implementation of online economic applications under XWiki. XWiki is a second generation Wiki system with all main features from Web 2.0. We take as an example an application that provides online audit and econometric calculations over the user's data on a specific enterprise (micro-economic analysis of gender equality on labour market). Calculations include data grouping, statistics, extended Mincer's equation, Duncan index of dissimilarity, the Oaxaca-Blinder decomposition.*

***Key words:*** *online economic applications, Wiki system, online audit, econometric calculations, gender equality on labour market.*

***JEL:*** *C02, C80, C88, C89*

## 1. Introduction: what is Web 2.0?

Concept of Web 2.0 is informal. It is the next stage of development of the Internet. Its main characteristics are replacement of static Web pages by pages with dynamic or user-generated content, and quick growth of social media. More detailed descriptions include also ease of use, participatory culture and interoperability for end users. Therefore, terms "Social Web" or "Participatory Web" are applied.

The term "Web 2.0" is known since 1999 and became popular since 2004.

Existence of Web 2.0 doesn't imply that all pages in World Wide Web are of the second generation. A lot of old-fashioned static pages continue to live and are developed anew.

[1] PhD. in Computer Science, "Vladimir Andrunachievici" Institute of Mathematics and Computer Science, e-mail:acolesnicov@gmx.com

[2] Scientific researcher, "Vladimir Andrunachievici" Institute of Mathematics and Computer Science, e-mail: lmalahov@gmail.com

[3] *This paper was elaborated in the framework of the scientific project for the years 2018–2020, entered in the State Register of Science and Innovation Projects with the code 18.80013.0807.06.STCU / 6336 "Innovative approaches to applied computations and software development for gender equality regulation on labor market" („Abordări novaţionale privind dezvoltarea calculelor aplicative şi software-ului pentru reglementarea egalităţii de gen pe piaţa forţei de muncă"), within the framework of the STCU-ASM Joint Research-Development Initiatives Program, the call from 10.04.2017.*

In fact, Web 2.0 cover more than Web pages; it reorganizes the whole paradigm and approaches to network interaction.

Let us take a task of file hosting as an example. Web 1.0 solutions like Akamai[4] or Letitbit[5] are widely known and used. Web 2.0 replacement for file hosting is *torrent*, where endusers distribute stored files directly from their computers without intermediary host and formal procedures, and even don't use fixed connection. In torrent, files are transferred by chunks of standard length; any chunk can be taken from any computer that keeps this chunk. It is possible that a computer gets chunks of a file and distributes in parallel other chunks of the same file, which this computer obtained before. The whole process is regulated automatically. Modern conception of cloud seems to be further development of Web 1.0 file hosting that can be positioned in between of old variant with its formality and regulation, and the anarchy of torrent.

Interoperability means great compatibility with other systems. This is another important characteristic of Web 2.0 as it plays the main role in the subject of this paper.

## 2. The developed software

We develop a Web application that helps companies and enterprises to estimate gender misbalance at the company employees, and helps specialists to provide recommendation on equalizing this misbalance, with possible generalization to the whole labour market.

The application supposes four types of users: visitors (guests), clients, specialists, and system administrators.

Visitor's access rights are restricted by main page and several pages with documentation. Visitor is not registered. He cant't start calculations but can see examples of those or video clips. To use other possibilities, visitor can apply for registration.

Client is a registered user that can present data of his company, perform calculations and get recommendations of specialists. His access rights cover additional pages where he stores his data, starts calculations, get results and recommendations, and discusses them. Each client works on separate pages because his presented data are kept as strictly confidential and can't be accessed by another client.

Specialist is a registered user that has access to data of all clients or of a group of clients, and to results of calculations. His task is economic analysis of clients' results and development of recommendations. A specialist can use client's data, for example, for his own scientific work, but only if it is explicitly permitted by the client. The client can put restrictions on usage of his data, for example "only generalized (total and subtotal) indices".

System administrator has full access to the whole application and data and perform support of its work.

[4] https://www.akamai.com/

[5] http://wm.letitbit.net/

## 3. Client's working session

Client can present data in several formats including CSV and Excel worksheet.

The data should contain indicators of all employees of the company or its subdivision for a year. The indicators are: gender, age, position at the company, level of education, working experience (in general, and at the company), marital status, number of children, sum of all payments, number of days for business trips, trainings, and illness, type of working conditions, etc. These data are necessary to estimate local gender misbalance of labour force.

For data, the second level of confidentiality is supported. The data should not contain any real personal information like name, SSN or passport number. Employee's string in the data table should be identified by a depersonalized code that exists only for these data and should not be induced from any personal data. We are not interested in personalizing the data because all calculations produce more or less generalized results, and only they will be used in research. Nevertheless, client can restrict usage of his data and results. Raw (source) client's data couldn't be transmitted to any third part in any case.

After the data are entered, the system verifies them and ask corrections if necessary.

Each data item should be validated individually. Additionally, there are dependencies between data items that can be used for their validation. As a general rule, all individual data verification is performed before their validation by dependencies.

Moreover, some data may be not presented. We differ mandatory and optional data. In some cases a lost item may be replaced by a default value.

We also differ determined and fuzzy dependencies.

At the moment we use 25 parameters collected in user's data tables, and three additional parameters listed below:

$a_{\min}$ – minimal age of labour permitted by the law, for example, $a_{\min}$=14

$a_{\max}$ – estimated maximal age, for example, $a_{\max}$=150

$a_{\mathrm{ad}}$ – adult age by the law, for example, $a_{\mathrm{ad}}$=18

Data verification module validates data applying the following algorithm.

1. Values to additional parameters depending on the country are set.
2. Presence of mandatory parameters is checked, and lost values are filled in by default values.
3. Individual validation of each parameter is performed, including data type correspondence.
4. Determined dependencies are checked.
5. Fuzzy dependencies are checked.
6. If a value is found invalid, the message is issued, and the check continues.

As the data seem to be correct, the client initializes prescribed calculations. Calculations include 16 algorithms that are divided in several groups.

A. Statistical analysis of some parameter from input data in general and by gender, in numbers and percent rates. There are 11 calculations of the kind, for:

gender distribution; level of education; age groups; marital status; number of children; work experience, total and in the area; work arrangements; territorial distribution (if applied); dismissals in total and by dismissal cause; illness, business trips, and trainings.

B. Analysis of salary distribution by gender, in percent rates.

C. Complex analyze of types of activity in general and by gender, also with additional grouping by salary, education, experience and working arrangements, plus average age and number of children, and marital status (%), in numbers and percent rates.

D. Mincer regression, with logarithm of monthly salary as the dependent variable.

E. Blinder-Oaxaca decomposition of gender difference in wages (salary) into explained and unexplained parts.

F. Duncan index of vertical segregation by gender.

The client gets and stores results of calculations. Results are shown in absolute quantities and in percent, and presented as tables and graphics (diagrams).Pandas library of Data Analysis exports data from Python DataFrame object in several formats, including HTML table, CSV (Comma-Separated Values), Excel spreadsheet, etc. HTML output is directly inserted into the XWiki dynamic HTML page.

For graphic presentation, in addition to XWiki extensions, we used the Python 3 library Matplotlib.

Specialists analyze these results and contact the client for discussion and recommendations.

This cycle may be repeated with slightly corrected data, or data for another year, or data for another subdivision of the company. Client is provided by the necessary space (pages) to store his data and results, and he can revise them as necessary, or copy for his own use.

## 4. Implementation of the software in the Web 2.0 environment

The gender audit platform is divided into 4 main functional modules: data import module, data verification module, data processing module, and results presentation module.

For programming, the XWiki[6] environment is used. Motivation for this selection was published in [1].

Additionally, more convenient toolsto program complex economic calculations are used, namely, Python 3 [2,3] with the Pandas [4,5] library (Python Data Analysis) and other necessary modules. The ability to use the necessary additional development tools is one of the power features of the Web 2.0 approach implemented in the XWiki environment.

The XWiki environment offers a number of extensions, some of which we use. However, this functionality is not always enough to implement the challenges.

[6] https://www.xwiki.org/xwiki/bin/view/Main/WebHome

The XWiki environment provides, as an extension, a macro to include a Python program directly in the code of a dynamic Web page. We tried this extension and found out that its functionality is not complete. Namely, the specified extension is implemented using Jython libraries that implement Python in the Java environment. With this approach, translation from the Python language is performed not into the Python interpreter codes, but into the codes of the Java virtual machine.Their execution is performed in the Java environment. Two restrictions follow from this: on the version of the language and on plug-in library technologies.

Jython implements an older version of Python 2, version 2.7, on which development of Python 2 was stopped. All innovations from Python 3 are thus unavailable.

In addition, only those Python libraries that use only the binary code of the Python interpreter, which can be easily translated into Java virtual machine code, can be transferred under Jython. However, most modern Python libraries make extensive use of machine code for optimization purposes. Such libraries, in principle, cannot be connected withJython. Unfortunately, this concerns just the most needed libraries that implement complex calculations, for example, the least squares method. These include the NumPy, SciPy, Pandas libraries and several others. Finally, from July 2019, support of the Pandas library for Python 2 has been discontinued.

The flexibility of the Web 2.0 approach helped us in that case as well. The simplest calculations in Python 2 were included directly in the code of the dynamic page. For more complex calculations (regression analysis, etc.), a slightly more complex workaround was used. In the future, it is possible to implement this method as a new XWiki extension. Being a Web 2.0 environment, XWiki welcomes user extensions, but we have not set this goal yet.

In our case, we used the most modern standard implementation of Python 3, namely, CPythonthat is based on C. This implementation has access to a full set of libraries, loaded from the PyPy repository at necessity, and an update tool. This implementation must be specially installed on the XWiki server both under Linux (our case) and under Windows.

At the same time, we got the opportunity to debug Python programs independently outside of the XWiki environment, including in specialized shells, and even on other machines, which we widely used. All that was needed was to ensure that all machines on the developers and the XWiki server had the same set of Python libraries installed.

Next, the JPserve library was installed.It is an implementation of Python calling tools from a Java program. This library consists of a server written in Python and a client implemented in Java. Being a Python module, the server part is installed in the usual way: with the *pip* command from the PyPy repository. The client library is added (copied) to the root directory of the Tomcat Web application server along with other Java archives that collectively implement the XWiki environment, and thus become part of XWiki.

You can’t use the root user to start the Python server for security reasons. For this purpose, a separate user was specially created, in whose partition the server is started and all calculations are performed, and all Python programs and intermediate data, if any, are stored. Using a dedicated user makes it impossible to damage from Python 3 areas belonging to other Linux users in the case of a possible developer error. During development, we started the Python server manually in the background, leaving it working until the system reboot. In the future, this process is subject to automation.

To run the economic computing program in Python and include the result of the calculations in a dynamic XWiki page, the Groovy macro is included in the code of this page. Groovy is an extension for Java. In fact, in this way Java code is included in the page. In this case, everything displayed by the *print()* or *println()* commands is considered the XWiki code (extended HTML) and displayed by the browser. It is very important that there can be several such fragments in different programming languages on a page, and all of them use a common pool of variables.

In the Groovy code, a client program is called, one of the parameters of which is the name of the Python program to be executed. The Java client (that is, XWiki) sends a request to execute the program on the port allocated for this purpose to the server. A previously running Python server accepts the request and runs the specified program. The result must be assigned to the variable *_result_* and returned to the client (in the Java program) in standard JSON encoding. Turning a JSON string into XWiki variable values or into HTML code will be performed by a standard program available from Java.

We used the same machine as a client (XWiki) and as a server (Python), although these may be different machines. The port is set when the Python server starts (by default *8888*), and the Python server URL is set when the client requests it (by default *localhost*).

## 5. Conclusion

In our case, software development in Web 2.0 environment was extremely comfortable, mainly because of its great interoperability/ This permitted us to use powerful Python 3 libraries to implement necessary algorithms. The difficulty level of this implementation is comparable with direct calculations in Excel or other econometrical application.

## References

1. Colesnicov, L. Malahov, E. Lucasenco. On Implementation Platform for Gender-Oriented Econometrical Calculations, pp. 232–244. Proceeding of the International Symposium “Experience. Knowledge. Contemporary Challenges”, 3rd Edition “Romania in the Year of the Centenary. The European and global socio-economic context”. December 13th–14th, 2018, Editura “Artifex”, Bucharest, Romania. – 696 p
2. M. Jaworski, T. Ziadé. Expert Python Programming, 3rd Edition. Packt. April 2019. – 646 p.L. Ramalho.Fluent Python:Clear, Concise, and Effective Programming. O'Reilly Media.August 2015–792 p.
3. W. McKinney.Python for Data Analysis, 2nd Edition.O'Reilly Media, Inc.October 2017
4. K. Sheppard. Introduction to Python forEconometrics, Statistics and Data Analysis. 3rd Edition. University of Oxford. February 2018. – 427 p. (Electronic edition)