

## VIRTUAL LAB IN MODELICA FOR AIR POLLUTION CONTROL\*

C. Martín-Villalba<sup>1</sup>, M. E. Manzur<sup>2</sup>, A. Urquía<sup>1</sup>

<sup>1</sup>Universidad Nacional de Educación a Distancia (UNED), Spain

<sup>2</sup>Universidad Nacional de Tucumán, Argentina

### Abstract

Interactive is a Modelica library whose goal is to facilitate the implementation of virtual labs based on Modelica models quickly and with little effort. Modelica is a free object oriented modeling language. The implementation of a virtual lab for air pollution control developed using Interactive 2.0 is discussed in this manuscript. This virtual lab has been developed to explain the dispersion of pollutants into the atmosphere to undergraduate students of Environmental Chemistry of the Universidad Nacional de Tucumán (Argentina). Main aspects in the virtual lab development process are addressed in this discussion, including: 1) application of a systematic methodology to adapt any Modelica model into a description suitable for interactive simulation; 2) composition of the virtual lab view using Interactive. Additionally, the use of this virtual lab in the Environmental Chemistry course is discussed. Interactive is freely available at [www.euclides.dia.uned.es](http://www.euclides.dia.uned.es).

**Keywords:** *Modelica, virtual lab, air pollution.*

**Citation:** C. Martín-Villalba, M. E. Manzur & A. Urquía, "Virtual Lab in Modelica for Air Pollution Control," *Computer tools in education*, no. 1, pp. 5–15, 2018.

## 1. INTRODUCTION

Modelica [1] is a free object-oriented modeling language that facilitates the description of models in terms of hybrid differential algebraic equations (i.e., hybrid DAE). The mathematical models are described using equations, algorithms, and clauses to define the hybrid behavior, such as when and if clauses. The assignment of computational causality, and the symbolic manipulation and sorting of the model equations, are not made by the model developer. The modeling environment performs automatically these tasks. This feature greatly facilitates model development and reuse, reducing the time and effort required for completing simulation projects. Besides, Modelica facilitates structuring the modeling knowledge into model libraries, and defining new models by instantiating and connecting the library components. Model connection is performed on the basis of the energy-balance principle. A list of some of the Modelica environments can be found in [1]. Two well-known Modelica environments are Dymola and OpenModelica. Dymola is a commercial environment and OpenModelica is a free environment developed by the OpenModelica Consortium.

---

\*Work by C. Martín-Villalba and A. Urquía was supported by the Ministerio de Economía y Competitividad of Spain, grant DPI2013-42941-R and by the European Commission, Erasmus+ Project No. 573751-EPP-1-2016-1-DE-EPPKA2-CBHE-JP.

Virtual labs facilitate user's interactive actions on the models through the interactive user-to-model interface. This user-to-model interface, named virtual lab view, typically includes an animation of the model behavior and gadgets such as sliders, check-boxes and buttons to change the model variables, parameters, etc. During the virtual lab running, it is established a communication between the virtual lab view and the simulation of the mathematical model. The view changes according to the actual state of the simulation of the mathematical model, and the modifications performed by the users in the view can change the mathematical model.

Interactive [2] is a free Modelica library that facilitates the implementation of virtual labs based on Modelica models. Interactive Release 2, which can be run in both Dymola and OpenModelica, is described in [2, 3]. Using Interactive 2.0 in combination with OpenModelica, we have a free solution to develop interactive virtual lab based on complex Modelica models. The virtual lab developer only needs to use the Modelica language to describe the virtual-lab view, and to set-up the virtual lab. The Interactive library includes all the code required to generate the graphical user-to model interface (a C++ application) and to perform the communication between the model and the view (through TCP sockets).

The components of Interactive Release 2 have been programmed using Qt 4.8, Qwt 6.1, QwtPlot3D and VTK 6.1. The Qt library [4], programmed in C++ and free for non-commercial use, has been used to develop the C++ code corresponding to the windows and the interactive elements (i.e., sliders, check-boxes, etc.). The Qwt and QwtPlot3D libraries [5], which contain Qt widgets for technical applications, have been used to implement 2D and 3D plot widgets. The VTK library [6], written in C++ and open source, has been used for the rendering and the visualization.

We have developed a virtual lab for air pollution control using the Interactive library and the Modelica language. This virtual lab can be executed with Dymola and OpenModelica, and it is currently being used by students taking the course Environmental Chemistry of the degree in Chemistry of the Universidad Nacional de Tucumán (Argentina). It is aimed to analyze the pollutant concentration emitted from large industrial chimneys (i.e. stacks) into the landscape. The pollutant dynamics are determined by: 1) transport, 2) diffusion, 3) chemical transformation, and 4) ground deposition [7, 8]. The polluted air coming from the stack is called plume. The plume shape can be influenced by the meteorological conditions and emission factors such as gas temperature at the stack top, stack height, pollutant mass flow rate, or wind velocity.

This virtual lab may help students to get insight into the physics that governs the circulation of pollutants in the atmosphere and into the pollutants chemical behavior. The pollutants emitted through a stack source enter the atmosphere and can be transported or dispersed by meteorological conditions. The virtual lab model considers dilution and concentration processes. Students can analyze by using this virtual lab the relationship between the shape of the plume, shown in the virtual lab main window, and the meteorological conditions. Students can set different meteorological conditions, wind speed, emission rate, terrain complexity and stack height, and then visualize the resulting plume shape and analyze the pollutant concentrations.

There are many models of air quality that consider the processes governing the contaminants dispersion and the reactions in the atmosphere [9]. Modeling the atmospheric dispersion is essential to calculate the pollutant concentration at specific points and to analyze how this pollution moves in the atmosphere [10]. These models are used to estimate the pollutants concentration at specified ground-level receptors surrounding an emission source. There are many atmospheric dispersion simulators provided and approved by US Environmental Protection Agency [11]. These simulators typically have as input information about emissions and meteorology and predict pollution concentration at several locations using a dispersion model. Our virtual lab is based on a Gaussian dispersion model.

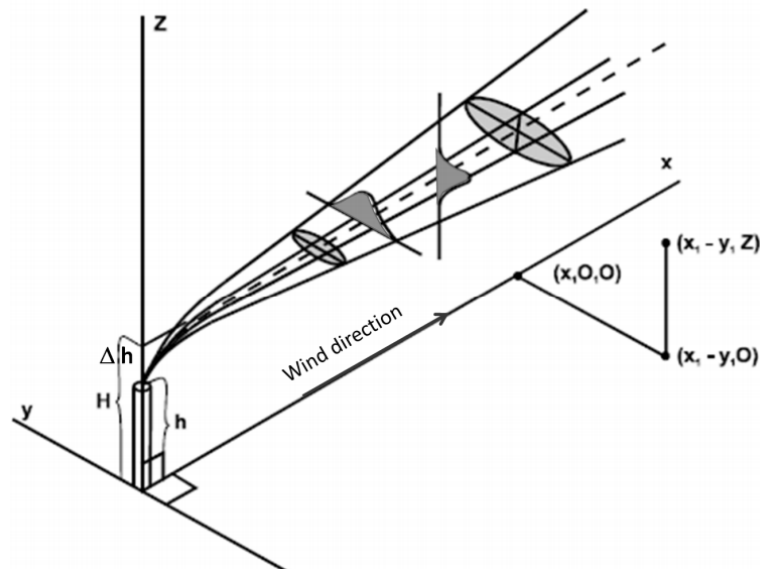


Figure 1. Gaussian plume model

This paper has the following structure. Firstly, the methodology to describe the virtual lab for the air pollution control using Interactive 2.0 is described. Then, the Modelica dispersion model and the steps to implement the user interface are discussed. Finally, the use of the virtual lab is explained.

## 2. METHODOLOGY

To implement the complete virtual lab in Modelica it is required to describe: (1) the Modelica model, (2) the view, and (3) the connection between the model and the view. The description of these three elements for the air pollution virtual lab is discussed below.

### 2.1. Description of the air pollution model

The Gaussian plume model is a standard approach for studying the transport of atmospheric pollutants [12, 13]. The main assumptions taken to develop this model are the following [14]:

- Steady-state conditions, which imply that the emission rate of the source is constant.
- Dispersion is negligible in the downwind direction.
- The wind speed, wind direction, temperature and mixing height are constant.
- No wind shear in the horizontal or vertical plane.
- The pollutants are non-reactive gases or aerosol.
- There is a perfect reflection of the plume at the underlying surface with no deposition or reaction with the surface.

Pollutant distributions are represented by concentrations on a (regular) three-dimensional grid of points. An orthogonal Cartesian reference system is assumed with its origin corresponding to the base position of the source. The  $x$  axis is parallel to the wind direction and the  $z$  axis corresponds to the height from the ground direction.

The Gaussian's method employs the Gaussian's normal distribution equation to determine the variation of the pollutant concentrations in the plume by relatively simple calculations. This

model has a symmetry axis determined by wind direction, as is illustrated in Figure 1. According to the Gaussian dispersion model, there exists a region of more pollutant concentration and two symmetrical regions. The pollutant concentration in the air in the coordinate  $(x, y, z)$  is described by the following equation:

$$C(x, y, z) = \frac{Q}{2 \cdot \pi \cdot u \cdot \sigma_y \cdot \sigma_z} \cdot \exp\left(-\frac{y^2}{2 \cdot \sigma_y^2}\right) \left\{ \exp\left[-\frac{1}{2} \left(\frac{z-H}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2} \left(\frac{z+H}{\sigma_z}\right)^2\right] \right\} \quad (1)$$

where  $C(x, y, z)$  is the concentration of pollutant in the air in  $\text{g/m}^3$ ,  $Q$  is the emission rate of the pollutant from the source in  $\text{g/s}$ ,  $\sigma_y$  is the horizontal crosswind dispersion coefficient,  $\sigma_z$  is the vertical dispersion coefficient,  $u$  is the mean speed of the wind in  $\text{m/s}$ , and  $H$  is the effective height of the plume in  $\text{m}$ .  $H$  is computed by adding the additional height to which the hot gases rise above the stack ( $\Delta h$ ) to the physical height of the stack ( $h$ ).

The following equation has to be applied to compute the concentration of a particular component in parts per million (ppm):

$$C_{\text{component}} = \frac{C \cdot 10^6 \cdot K}{MW} \quad (2)$$

where  $C_{\text{component}}$  is the component concentration in ppm,  $C$  is the concentration on pollutant in the air in  $\text{g/m}^3$ ,  $K$  is a constant equal to 24.5 at 1 atm and  $25^\circ \text{C}$ , and  $MW$  is the molecular weight of the component expressed in  $\text{g/mol}$ .

The value of the horizontal and vertical dispersion coefficients ( $\sigma_y$  and  $\sigma_z$ ) depends on the distance from the source and the atmospheric conditions. They can be calculated with different theories: Martin, Briggs, Pasquill, Taylor, etc. We have computed these coefficients using the equations shown in [15]. Pasquill defined the following seven classes of atmospheric stability conditions: A, B, C, D, E and F. The parameters correspond to “very stable” (F class) to “very unstable” (A class). The neutral atmosphere is represented by D. The Martin equations [15] to compute the dispersion coefficients are the following:

$$\sigma_y = a \cdot x^b \quad (3)$$

$$\sigma_z = c \cdot x^d + f \quad (4)$$

where  $x$  is the distance to the source expressed in  $\text{km}$ , and  $a$ ,  $c$ ,  $d$  and  $f$  are stability-dependent constants, and  $b$  is a constant of value 0.894. The  $c$ ,  $d$  and  $f$  constants assume different values depending on  $x$  is  $\leq$  or  $>$  1  $\text{km}$ . The value of these constants are shown in Table 1.

We suppose that it is only known the wind speed at a height of 10  $\text{m}$  ( $u_{10}$ ) and we compute the mean wind from Eq. (5), where  $p$  is a coefficient that depends on the stability class and the media (see Table 2).

$$u = u_{10} \cdot \left(\frac{h}{10}\right)^p \quad (5)$$

The Modelica model has been adapted to suit interactive simulation. A systematic methodology for transforming any Modelica model into a description suitable for interactive simulation was proposed in [16]. Basically, the interactive quantities need to be defined as state variables of the interactive model. For instance, the interactive parameters are defined as state variables with zero time derivative.

**Table 1.**  $a$ ,  $c$ ,  $d$  and  $f$  coefficients.

Stability class	$a$	$c$		$d$		$f$	
		$x \leq 1 km$	$x > 1 km$	$x \leq 1 km$	$x > 1 km$	$x \leq 1 km$	$x > 1 km$
A	213	440.8	459.7	1.941	2.094	9.27	-9.6
B	156	106.6	108.2	1.149	1.98	3.3	2.0
C	104	61	61	0.911	0.911	0	0
D	68	33.2	44.5	0.725	0.516	-1.7	-13.0
E	50.5	22.8	55.4	0.678	0.305	-1.3	-34.0
F	34	14.35	62.6	0.74	0.18	-0.35	-48.6

**Table 2.**  $p$  coefficient.

Stability class	$p$ in rural media	$p$ in urban media
A, B	0.07	0.15
C	0.10	0.20
D	0.15	0.25
E	0.35	0.40
F	0.55	0.60

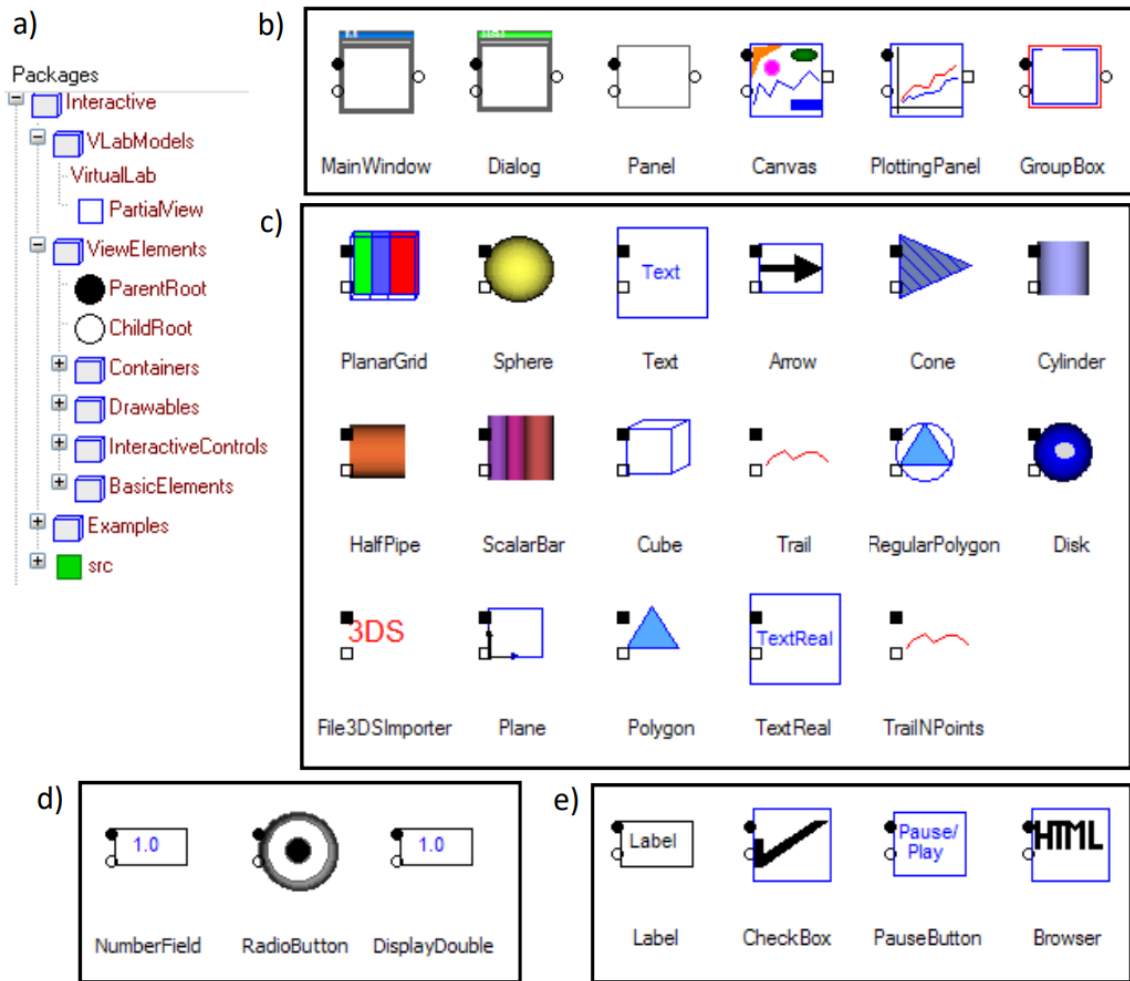
## 2.2. Description of the virtual lab view

The Modelica description of the view has been composed using components from the Interactive library. The Interactive library is structured in four packages (see Fig. 2a). *VLabModels*, *ViewElements* and *Examples* contain the Modelica classes that a library user (i.e., the virtual lab developer) will typically employ for developing virtual labs. On the contrary, *src* contains partial classes and Modelica functions that are not intended to be directly used by the virtual lab developers.

The *VLabModels* package contains classes that need to be used in the virtual lab view definition and for setting up the complete virtual lab. We have created a new class, named *PollutantsView*, by extending the *PartialView* class. The *PartialView* class includes the root class, a graphic component that hosts the rest of the graphic components, and the code to perform the communication between the model and the view. The required graphic components are dragged and dropped with the mouse from the *ViewElements* package to the *PollutantsView* class. Then, they are connected forming a tree structure (see Figure 3). The connections among the graphic components determine their layout in the virtual lab view.

The Modelica description of the graphical interface of this virtual lab is depicted in Figure 3. The graphical user interface generated is shown in Figure 4. The Modelica description of the view (see Figure 3) has a main window (*MainWindow* class) that hosts a panel of the *Panel* class on the top, a container of the *Canvas* class at the center, and another panel of the *Panel* class at the bottom. These classes are located inside the *Container* package. The graphical properties of these elements are set in the view definition and they cannot be changed during the simulation run.

The top panel includes objects of the *BasicElements* and *InteractiveControl* package that facilitate the user to interact with the model. The model variables can be linked to the state of the graphical components included in the *InteractiveControl* package. As a result, the value of the linked variable changes accordingly to the interactive changes that the virtual lab user performs on the control element during the simulation run. The top panel contains numerical boxes to set the wind speed ( $u$ ), the emission flow rate of the stack ( $Q$ ), the vertical plume rise ( $\Delta h$ ) and the



**Figure 2.** Interactive Release 2: a) general structure; and classes within the following packages: b) Containers; c) Drawables; d) InteractiveControls; and e) BasicElements

stack height ( $h$ ), and the  $x$ ,  $y$  and  $z$  coordinates of the trial point (i.e., a ball inside the volume whose pollutant concentration is shown). Additionally, there are radio-boxes to select the type of pollutant (sulfur dioxide or carbon monoxide), the class of stability, and the type of terrain among five different types (three of rural and two of urban type).

The object of Canvas class include objects of the *Drawable* package depicting the stack and the pollutant distribution. The geometric and visual properties of these objects (e.g., spatial position, rotation, size, texture and color) can be linked to the model variables, changing during the simulation according to the time evolution of these variables. The stack and the pollutant distribution are represented by an object of the Cylinder and the PlanarGrid classes, respectively. The pollutant distribution is represented as a spatial grid of equal size cuboid cells. Each cell has a different color depending on the pollutant concentration at the center of the cell. An object of the ScalarBar class indicates the relationship between the cell color and its pollutant concentration.

The bottom panel contains two check-boxes. A check-box that allows to display a dialog window with a plot of the pollution concentration at a determined height ( $z$ ) in the plane ( $x, y$ ). The other check-box allows to display a dialog window with the virtual lab documentation in HTML format.

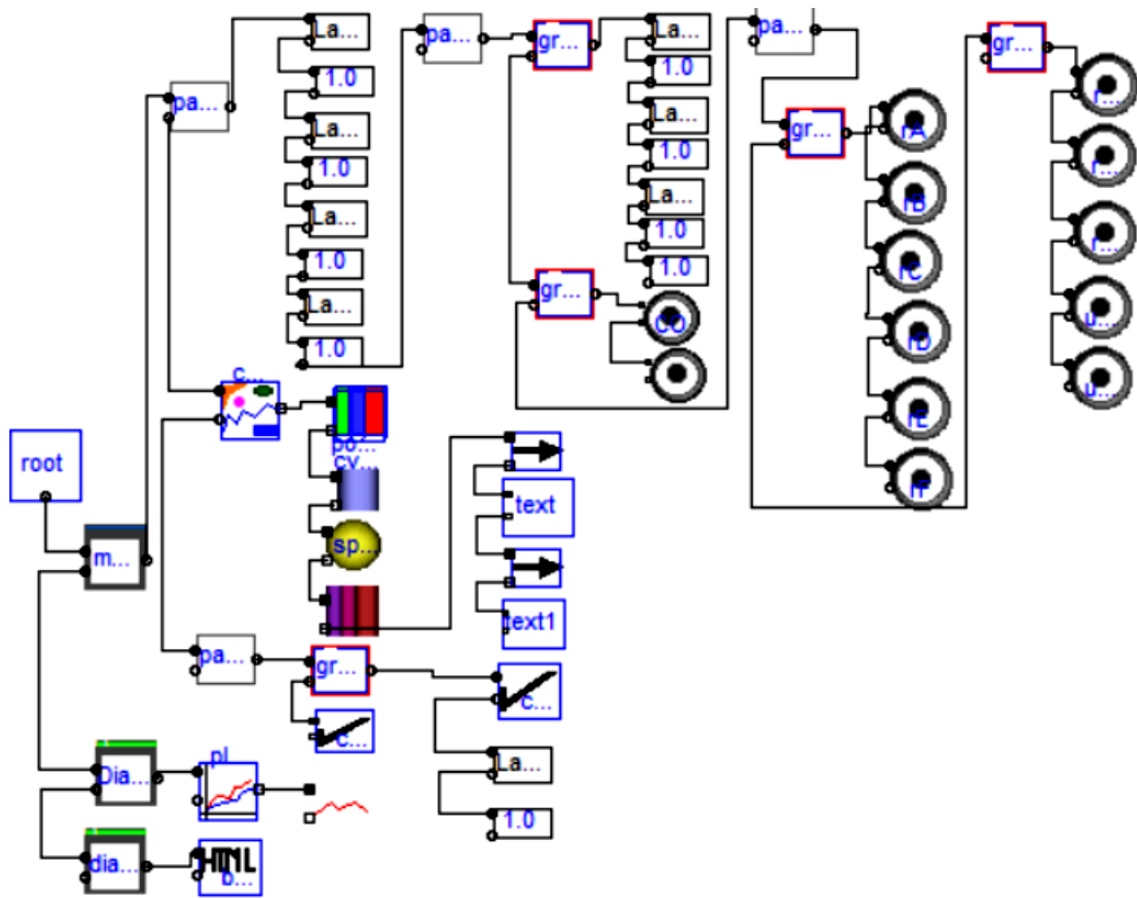


Figure 3. Modelica description of the virtual lab view

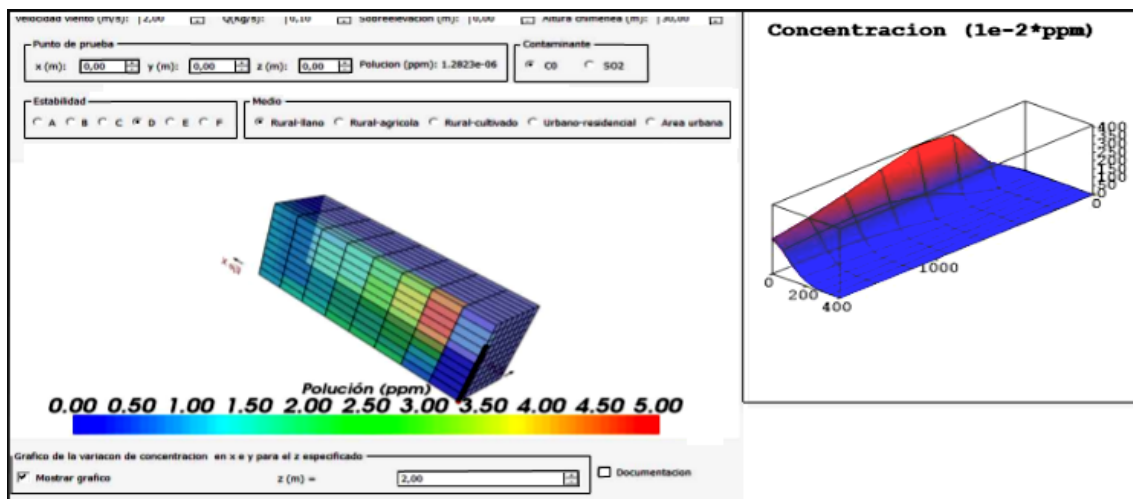


Figure 4. Virtual lab view



### 2.3. Description of the connection between the model and the view

We have created a Modelica class named VLab and we have inserted in it an instance of VirtualLab class, which is included in the *VLabModels* package of the Interactive library. VirtualLab contains two parameterized generic classes: the classes of the virtual lab model and view. We have written inside the VLab class equations to connect the model and the view variables.

The VLab class is translated using Dymola or OpenModelica, and run. As a part of the model initialization (i.e., the calculations performed to find the initial value of the model variables), the initial sections of the interactive graphic objects and of the PartialView class are executed. These initial sections contain calls to Modelica functions, which encapsulate calls to external C-functions. These C-functions generate the C++ code of the view. As a result, during the model initialization, the C++ code of the virtual lab view is automatically generated. Also, the communication procedure between the model and the view is set up. This communication is based on client-server architecture: the C-program generated by the simulation environment is the server and the C++ program (which is automatically generated during the model initialization) is the client.

During the simulation run, there is a bi-directional flow of information between the model and the view. The communication is as follows. Every communication interval, the model simulation (i.e., the server) sends to the view (i.e., the client) the data required to refresh the view; and the view sends to the model simulation the new value of the variables modified due to the user's interactive action.

## 3. DISCUSSION

This virtual lab is aimed to determine the concentration of pollutants (such as sulfur dioxide and carbon monoxide) for particular conditions of wind speed ( $u$ ), stack height ( $h$ ) and stability class ( $A, B, C, D, E, F$ ) in different terrains. Additionally, students can select the stability class and calculate the pollutant maximum concentration at downwind direction. The effect of the stability class can also be analyzed for different wind speeds ( $u$ ). The purpose is not the accurate prediction of the pollutants concentration from the site of generation to a site of deposition based on emissions and meteorological data, but to understand the complex relationships among them.

Students taking the Environmental Chemistry course are asked to perform the following activities using this virtual lab and a document with the maximum carbon monoxide and sulfur dioxide allowed during a certain time by the argentinian air quality legislation.

1. Obtain the carbon monoxide and sulfur dioxide concentration values emitted by a stack with effective height of the plume ( $H$ ) equal to 30 m at  $x = 100$  m, changing the following set of conditions: stability class, wind speed ( $u = 2, 4, 7$  m/s), emission rate ( $Q = 50, 100$  g/s), horizontal position ( $y = 25, 75, 125$  m) and vertical position ( $z = 2, 10, 18$  m).
2. Write a table with the value of the maximum concentration of pollutant and the point ( $x, y$ ) of this maximum, for  $z = 2$  m and for each stability class. Use for this activity the concentration plot ( $10^{-2}$  ppm).
3. Compare the values obtained in the previous activity with the maximum values permitted by the Argentinian quality air legislation for atmospheric pollutants. Are the maximum values exceeded in any of the cases?
4. Get the minimum value of  $Q$  that makes the maximum concentration of CO to be higher than the maximum permitted by the Argentinian legislation.



#### 4. CONCLUSIONS

This virtual lab can be used as an introduction to mass transport in environmental flows providing an introduction to the physics and chemistry of the atmosphere. Furthermore, this tool allows students to take account of the critical situation of their surrounding and to be strongly involved in sustainable development.

The implementation of this virtual lab is based on the application of a systematic methodology to transform any Modelica model into a formulation suitable for interactive simulation; the Interactive Modelica library, which allows performing the object-oriented description of the virtual lab view and the set-up of the complete virtual lab; and Dymola or OpenModelica, which translates the Modelica description of the virtual lab into an executable file. The use of Modelica reduces considerably the modeling effort, facilitating the code reuse and the development of virtual labs based on complex models. The simulation code generated by Dymola and OpenModelica is highly efficient. Interactive Release 2 is backward compatible, can be run in both Dymola 2015 and OpenModelica, and is freely available at [www.euclides.dia.uned.es](http://www.euclides.dia.uned.es).

#### References

1. Modelica and the Modelica Association. URL: <https://www.modelica.org/> (accessed february 2018).
2. C. Martín-Villalba & A. Urquía, “New features in the Interactive Modelica library”, *Humanities and Science University Journal of the Saint Petersburg University Consortium*, vol. 22, pp. 37–48, 2016.
3. C. Martín-Villalba, A. Urquía, Y. Senichenkov & Y. Kolesov, “Two approaches to facilitate virtual lab implementation”, *Computing in Science and Engineering*, January/February 2014, pp. 78–86, 2014.
4. Qt Website. URL: <https://www.qt.io/> (accessed february 2018).
5. Qwt Website. URL: <http://qwt.sourceforge.net/> (accessed february 2018).
6. VTK Website. URL: <https://www.vtk.org/> (accessed february 2018).
7. D. Alemayehu & F. Hackett, “Gaussian Dispersion Model to Estimate the Dispersion of Particulate Matters (Pm<sub>2.5</sub>) and Sulfur Dioxide (SO<sub>2</sub>) Concentrations on Tribal Land, Oklahoma”, *American Journal of Environmental Sciences*, vol. 11 (6), pp. 440–449. 2015.
8. D. Vallero, *Fundamentals of Air Pollution* (5<sup>th</sup> Ed), pp. 683–753. 2014.
9. N. S. Holmes & L. Morawska. *Atmospheric Environment*, vol. 40 (30), pp. 5902–5928. 2006.
10. N. A. Mazzeo & L. E. Venegas. *Environmental Modelling and Assessment*. Vol. 13 (3), pp. 349–356. 2008.
11. US EPA Website. URL: <https://www.epa.gov/scram/air-quality-dispersion-modeling-alternative-models> (accessed february 2018).
12. A. Abdel-Rahman. “On the atmospheric dispersion and Gaussian plume model”, 2<sup>nd</sup> Intl. Conference on waste management, water-pollution air pollution, indoor climate. WWA’08.
13. M. E. Manzur, G. Benzal & S. N. González, *Simulación de dispersión de contaminantes atmosféricos*. Archivos de Bioquímica, Química y Farmacia. Universidad de Tucumán. T. XXII (N. 2).
14. S. Brusca, F. Famoso, R. Lanzafame, S. Mauro. A. Marino & P. Monforte, “Theoretical and experimental study of Gaussian Plume model in small scale system”, 71<sup>st</sup> Conference of the Italian Thermal Machines Engineering Association, ATI2016.
15. D. O. Martin, “The change of concentration standard deviations with distance”, *J. Air Pollution Control Assoc*, vol. 26 (2), pp. 145–147. 1976.
16. C. Martín-Villalba, A. Urquía & S. Dormido, “An approach to virtual-lab implementation using Modelica”, *Mathematical and Computer Modelling of Dynamical Systems*, vol. 14(4), pp. 344–360, 2008.

Received 10.01.2018, the final version — 15.02.2018.

Компьютерные инструменты в образовании, 2018

№ 1: 5–15

УДК: 004.4 : 504.3.054

<http://ipo.spb.ru/journal>

## ВИРТУАЛЬНАЯ ЛАБОРАТОРИЯ ДЛЯ КОНТРОЛЯ ЗА ЗАГРЯЗНЕНИЕМ ВОЗДУХА В MODELICA

К. Мартин-Вильяльба<sup>1</sup>, М. Е. Манзур<sup>2</sup>, А. Уркиа<sup>1</sup>

<sup>1</sup>Национальный университет дистанционного обучения, Мадрид, Испания

<sup>2</sup>Национальный университет Тукумана, Сан-Мигель-де-Тукуман, Аргентина

### Аннотация

В статье рассматривается вопрос о внедрении виртуальной лаборатории для контроля за загрязнением воздуха, разработанной с использованием Interactive 2.0. Interactive — это библиотека бесплатного объектно-ориентированного языка моделирования Modelica, целью которой является облегчение реализации виртуальных лабораторий, основанных на ее моделях. Рассматриваемая виртуальная лаборатория была разработана для объяснения дисперсии загрязняющих веществ в атмосфере для студентов факультета Экологической химии Национального университета Тукумана (Аргентина). В статье рассматриваются основные аспекты процесса разработки виртуальной лаборатории, включая: 1) применение систематической методологии для адаптации любой модели Modelica для интерактивного моделирования; 2) использование Interactive для знакомства с виртуальной лабораторией. Обсуждается использование этой виртуальной лаборатории в курсе Экологической химии. Interactive находится в свободном доступе по адресу [www.euclides.dia.uned.es](http://www.euclides.dia.uned.es).

**Ключевые слова:** *Modelica, виртуальная лаборатория, загрязнение воздуха.*

**Цитирование:** Martín-Villalba C., Manzur M. E., Urquía A. Virtual Lab in Modelica for Air Pollution Control // Компьютерные инструменты в образовании. 2018. № 1. С. 5–15.

*Поступила в редакцию 10.01.2018, окончательный вариант — 15.02.2018.*

**Карла Мартин-Вильяльба, Факультет информатики и автоматике, Национальный университет дистанционного обучения, Мадрид, Испания, [carla@dia.uned.es](mailto:carla@dia.uned.es)**

**Мария Евгения Манзур, Кафедра экологической химии, Факультет биохимии, химии и фармации, Национальный университет Тукумана, Сан-Мигель-де-Тукуман, Аргентина, [memanzur@fbqf.unt.edu.ar](mailto:memanzur@fbqf.unt.edu.ar)**

**Альфонсо Уркиа, Факультет информатики и автоматике, Национальный университет дистанционного обучения, Мадрид, Испания, [aurquia@dia.uned.es](mailto:aurquia@dia.uned.es)**

**Carla Martín-Villalba,**  
Departamento de Informática y Automática,  
Universidad Nacional de Educación a Distancia  
(UNED); Juan del Rosal 16, 28040 Madrid, Spain  
[carla@dia.uned.es](mailto:carla@dia.uned.es)

**María Eugenia Manzur,**  
Cátedra de Química Ambiental. Facultad de  
Bioquímica, Química y Farmacia. Universidad  
Nacional de Tucumán, Tucumán, Argentina  
[memanzur@fbqf.unt.edu.ar](mailto:memanzur@fbqf.unt.edu.ar)

**Alfonso Urquía,**  
Departamento de Informática y Automática,  
Universidad Nacional de Educación a Distancia  
(UNED), Juan del Rosal 16, 28040 Madrid, Spain  
[aurquia@dia.uned.es](mailto:aurquia@dia.uned.es)

©

Our authors, 2018.  
Наши авторы, 2018.