EM WORKSHOP : AN INTERACTIVE TOOL TO BUILD WEB-BASED ELECTROMAGNETICS COURSES

JEAN-MICHEL DRICOT

and

PHILIPPE DE DONCKER

Département d'Electricité Générale
Université Libre de Bruxelles
Elecgen CP165/51
Avenue Roosevelt, 50
1050 Bruxelles, Belgium
E-mail : jdricot@ulb.ac.be, pdedonck@ulb.ac.be

Abstract

A software called EM Workshop has been developed to make easier the creation of interactive illustrations for web-based electromagnetics courses. The Workshop uses a high-level object-oriented programming model so that it can be easily adapted or complemented by the user. The definitive goal of EM Workshop is the availability for the educational community of a tool which allows the creation of interactive lessons without any knowledge in graphical interfaces programming.

KEYWORDS : electromagnetics, Internet, Java

Introduction

In the next few years, computer based courses will become common tools to complement or even to replace [1] the classical courses, namely concerning electromagnetics, a field of physics where the students have major difficulties to visualize even the most simple phenomena. The main advantage of computer based lessons is coming from the fact that the students can interact with the figures illustrating the theory to obtain an "instant feedback" allowing a more intuitive knowledge of the physical systems.

The idea of implementing on computers teaching tools in electromagnetics is not new. Several Universities have already developed demonstration softwares based either on commercial scientific computing codes [2] or on their own interfaces [3,4]. However, the huge development of the Internet provides new capabilities which seemed totally impossible a few years ago [5]. The implementation of lessons on the Internet thanks to the Java language gives the possibility to the students to dispose of a learning support whenever and wherever they want, for example at home, and on any computer where an Internet browser has been installed. Furthermore, since the lessons are stored on a single server, they can be easily updated.

However, the development of interactive graphical interfaces on the Internet with Java is still a very tedious task and the programmer's background necessary to build such interfaces is not yet widespread in the academic world. To circumvent this problem a tool called *EM Workshop* has been developed to build interactive figures as easily as non-interactive figures can be built with the common commercial drawing softwares. In EM Workshop, each physical entity (charge, current, antenna,..) constitutes an object in the sense of the object-oriented languages containing its definition, its graphical representation and its electromagnetic behavior. A figure can be built by simple drag-and-drops with the mouse of these objects. When the figure has been drawn, EM Workshop creates the corresponding Java applet displaying interactively the different objects and computing the associated electro-magnetic fields or potentials.

The Java Language

Java [6] is two things: a *programming language* and a *platform*. It provides simple, robust, object-oriented and architecture-neutral development possibilities and high-performance and backward compatible (mostly with C and FORTRAN) codes.

Java is unusual because each Java program is both compiled and interpreted. With a compiler, the Java program is translated into an intermediate language called Java bytecodes : the platform-independent codes interpreted by the Java interpreter. With an interpreter, each Java bytecode instruction is parsed and run on the computer. Compilation happens just once, interpretation occurs each time the program is executed. Figure 1 illustrates how Java works. As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecodes compilers can bring Java's performance close to that of native code without threatening portability.

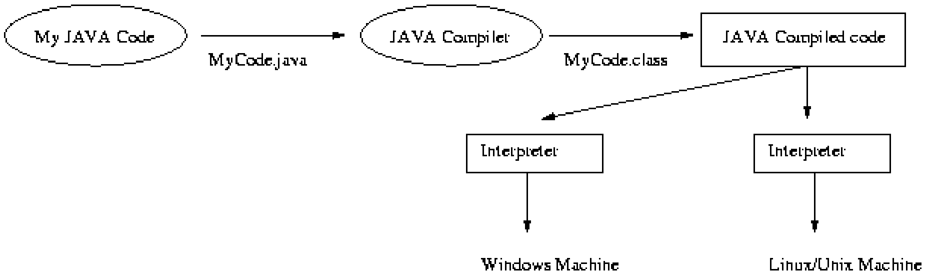

Figure 1: The Java language

You can think of Java bytecodes as the machine code instructions for the *Java Virtual Machine (Java VM)*. Any Java interpreter, whether it is a Java development tool or a Web browser which can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java bytecodes make *"write once, run anywhere"* possible. A Java program can be compiled into bytecodes on any platform having a Java compiler. The bytecodes can then be run on any implementation of the Java VM: the same Java program can run on Windows, Solaris, Linux and Macintosh.

The Object Oriented Programming Model for Physics

All programming languages provide abstractions. It can be argued that the complexity of the problems you can solve using a language is directly related to the kind and quality of its abstraction. Assembly language was the first step in the abstraction of the underlying machine. Next, many so-called "imperative" languages that followed (such as FORTRAN, BASIC, and C) were abstractions of the assembly language providing many improvements. However, their abstraction level still required to think in terms of the structure of the computer rather than in terms of the structure of the problem being solved.

The object-oriented approach takes a step further by providing tools to represent elements in the problem space. This representation is general enough to not constrain the programmer into any particular type of problem. It is not only a new way of thinking a problem, it is also a more natural, conceptual way of implementing it. It is easy to see how Object Oriented Programming (OOP) can change the way of thinking computational physics on the simple problem of the computation of the electric field due to an assembly of point charges located in a given region of space. The problem can be faced using classical implementation by creating small routines to compute the electric field of a given number of point charges whose positions and values are stored in an array. In OOP, the problem is thought more conceptually. Two objects are designed: the “charge” object and the “region of space“ object, having each their own attributes and their own behavior. Then the interactions between the objects are implemented and the problem is modelized: an assembly of charge objects is “attached” to the region of space object which is able to compute the electric field throughout itself by “asking” the charges their positions and the individual fields they produce.

In OOP, the term “object” thus refers to any physical entity of a problem. But it is obvious that such objects can share common attributes or common way of interacting. They can be putted in a common "box" known as a *class* in Java. The concept that all the objects, while being unique, are also part of a set of objects that have characteristics and behaviors in common is an essential particularity of object-oriented languages. Solenoids and magnets have in common the fact they generate a magnetic field: generally speaking of these objects, they could be referred as being parts of a "magnetic" class.

The mechanism which allows objects to be "packed" in classes is called the "generalization mechanism". The reverse mechanism also exists and is called "specialization" or "inheritance". It allows objects to derivate from more general components. Using this mechanism, attributes and behavior functions (called *methods* in Java) of a *subclass* are directly inherited from its *superclass*.

EM Workshop: an Object Oriented Workshop for Electromagnetics

EM Workshop is an attempt to use high-level Object-Oriented language to modelize problems relative to electromagnetism. It provides:

1. deployment ease: from local to Internet usage
2. high portability and reusability of code
3. backward compatibility with older routines
4. fast development for teaching use

The main goal of this Workshop is to provide basic support and architecture for the creation of small and interactive illustrations of electromagnetism. Till now, the work space allows 2D representation of 3D problems. Unlike other end-oriented applications available on the World Wide Web, EM Workshop tends to be very abstract and *reusable* thanks to the use of the OOP model.

EM Workshop provides a global work interface and some basic features allowing developers to create new classes, based on existing ones in the workshop, with minimal coding time and with basic knowledge of the Java language. Workshop's components can also behave like Java graphical components and be included in other existing Java programs (more precisely: EMW components *are* Java graphical components).

EM Workshop presents two main parts: a Library containing all the classes and methods necessary to describe the physical objects and a Builder allowing the direct manipulation with the mouse of all the library objects to create interactive figures.

The Library structure is based on three kinds of components:

1. The "Universes" which are able to contain all the electromagnetic objects.
2. The "EMObjects" : the electromagnetic objects (charges, antennas, dielectric bodies..) to be placed in a Universe
3. The "Math Utils" : a set of computation routines for specific electromagnetic problems and of various mathematical routines coming mostly from LaPack and LinPack.

Till now, four main classes of electromagnetic objects exist: *ElectricObjects* for electrostatic, *MagneticObjects* for magnetostatic, *Antennas* and *Medium* for dielectric, magnetic or conducting structures. Each class defines set of properties and behaviors called an interface which must be implemented by any object to be compatible with the Workshop. For instance, any ElectricObject must be able to give the electric field it generates at a given location or its own position in the Universe.

The Universes contain the EMObjects and are responsible of the drawing of the objects and of the generated fields or potentials. They are helped in their task by various mathematical routines, mesh generators or user-defined computation classes.

The interactions between the objects, the graphical interface, and the mathematical resolution is entirely made inside the Library but the user can design its own objects or resolution routines (eventually in C or Fortran) and include them in the Library.

The second part of the Workshop is the Builder*:* an intuitive and visual development interface. It is not really necessary to learn about Java or OOP concepts to create a web-based lesson. Everything can be done interactively in the Builder using the objects already defined in the Library. For instance, to create a figure showing the electric field produced by two point charges, it is just necessary to take them in a toolbox and to put them into a Universe by simple drag-and-drop with the mouse. The Builder then creates the corresponding Java applet and once deployed on the Internet, the figure becomes fully interactive: the student can change the position or the value of the charges while the electric field is updated in real time.

Conclusion

Interactive courses for electromagnetics on the Internet constitute a very promising learning support. However, the development of such courses is still a very tedious task requiring high level skills in software design. To circumvent this problem a tool called EM Workshop has been built to create interactive illustrations of electromagnetism as easily as non-interactive figures can be drawn with classical commercial drawing softwares. By using an elaborate object-oriented structure, the workshop provides highly reusable Java classes and it is the authors' hope that it could be adapted in the future to other fields of physics.

The EM Workshop Library and EM Workshop Builder can be freely downloaded on the Internet at http://maxwell.ulb.ac.be/emworkshop. An application example of the Workshop concerning the propagation of electromagnetic waves through the human body can be found at http://maxwell.ulb.ac.be/adamproject

References

1. Coleman, D.J. Kinniment, F.P. Burns, T.J. Butler, A.M. Koelman, “Effectiveness of computer-aided learning as a direct replacement for lecturing in degree-level electronics”, *IEEE Trans. Education*, **41**, pp. 177-184 (1998)

2. F. Aubanel, M. Hélier, F. Jouvie, W. Tabbara, “Electromagnetica, a Mathematica based electromagnetic course”, *Proceedings PIERS’98*, pp. 1127, Nantes, France (1998)

3. F. Buret, D. Muller, “Computer-aided education for magnetostatics”, *IEEE Trans. Education***, 42,** pp. 45-49 (1999)

4. A.E. Elsherbeni, D. Kajfez, J.A. Hawkes, “Interactive display of vector fields inside waveguides”, *IEEE Trans. Education*, **36,** pp. 283-286 (1993)

5. R.W. Cole, “Web-based lessons, changing the face of electromagnetics education”, *Proceedings PIERS’98*, pp. 1122, Nantes, France (1998)

6. M. Campione, K. Walrath: The Java tutorial, object-oriented programming for the Internet, Addison-Wesley (1998)