



HAL
open science

Pliant: more than a programming language, a flexible e-learning tool.

Patrice Ossona de Mendez, Marcus Vicinius Santos, Isaac Woungang

► **To cite this version:**

Patrice Ossona de Mendez, Marcus Vicinius Santos, Isaac Woungang. Pliant: more than a programming language, a flexible e-learning tool.. 2004, pp.505-510. hal-00005643

HAL Id: hal-00005643

<https://hal.science/hal-00005643>

Submitted on 19 Aug 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pliant: More than a Programming Language, a Flexible E-learning Tool

Marcus Vinicius Santos, Isaac Woungang
Department of Mathematics, Physics, and Computer Science
Ryerson University, Toronto
Canada
m3santos@scs.ryerson.ca, iwoungan@scs.ryerson.ca

and

Patrice Ossona de Mendez
UMR8557, CNRS/EHESS, Paris
France
pom@ehess.fr

Abstract: The increasing importance of e-learning, and in particular distance learning, has been a boosting element for the emergence of Internet based education services. As we move into the information age, tremendous efforts are made in the design and development of new information and communication technologies for educational services, providing for a proliferation of new sophisticated e-learning management systems. A common performance metric of such a system can be considered to be its ability to meet the requirements of the providers of educational services, while integrating the advantages of human resource and economical limitations. However, under this feverish period of multimedia competition, the above performance measure should incorporate other technology deployment considerations such as software and hardware adaptation. Without such additional dimensions, current overlay e-learning systems software platforms would be limited only to services that they can accommodate. Consequently, the design and implementation of new software and hardware tools with "flexible" integration and encapsulation capabilities is always desirable. Pliant is a Web technology that can provide such potentials. This paper reports on the capabilities of Pliant. It shows how Pliant can provide a significant headway in resolving a number of common issues that are typically encountered with traditional approaches to Web-based technologies, e-Learning management systems, and related platforms development, while meeting educational and software-oriented expectations.

Introduction

It is likely that in a near future, the Internet, and the Web technologies underlying it, will be at least as pervasive, convenient, and invisible as electricity is today. To afford this "universal usability" (Hochheiser, 2001) of the Web, researchers have been working on the development of technologies to endow Web literacy to both young and old, novice and expert. Pliant is one of such technologies, but, with additional capabilities such as e-learning features. This paper reports on the capabilities of Pliant. It shows how Pliant can provide a significant headway in resolving a number of common issues that are typically encountered with traditional approaches to Web-based technologies, e-Learning management systems, and related Web applications development, while meeting educational and software-oriented expectations. This paper is structured as follows. In Section II, we present what Pliant is all about. Section III concentrates on general features of the

Pliant language, and the Pliant e-learning capabilities are described in Section IV. Finally, in Section V, we conclude our work and delineate Pliant's future developments.

What Pliant is All About

Hubert Tonneau initiated the Pliant project in 1984. He observed that the lack of coherence between applications, libraries, etc., often required a large amount of glue between relevant pieces of code. Also, it seemed impossible to conciliate high level constructions allowing improved expressiveness and conciseness in specific contexts, with low level adaptability allowing efficiency and optimized handling of exceptional cases. From these considerations, the necessity appeared to introduce a new efficient multi-level language having a flexible syntax and structure, which could be adapted to particular application contexts. The Pliant language is thus oriented toward efficiency, understood in terms of computational resources, as well as programming adaptability (De Mendez, 1999). The main structure of the language follows: modularity, dynamic compilation and a full reflexivity allowing the redefinition of the syntactical, compilation and code optimization rules.

New applications services have then been integrated at language level (good examples are scheduling primitives and database management) hence suppressing usual gaps and interfaces between applications. From this point of view, an application is seen as a set of libraries, or even as a language extension possibly introducing its own syntactical changes. These applications may also be gathered into a coherent execution context, leading to an actual operating system, which we shall call *Fullpliant* (which source code is 4.2Mb big only). This framework may actually be executed in two different ways: as a program executing various servers (on Linux or Windows platforms), or as an operating system lying alone on top of a Linux kernel. Available servers include DNS, FTP, POP3, SMTP, LPD, remote execution, secured channel and HTTP servers. The HTTP multi-site web server provides the standard application interface. That for, a powerful server-side dynamic page mechanism has been introduced, on which rely existing applications (Forum, Photography correction and high fidelity printing, Web-mail, etc.) as well as additional HTTP-related servers (like WebDAV). The limitation of the HTML/Javascript scheme should soon lead to the introduction of an enhanced extended Pliant browser valuable as a state of the art user interface for possibly distributed application software.

Being used in an industrial context since 2000, *Fullpliant* is also concerned with security issues. The transparent integration in the dynamic page extension of signature and right verification mechanisms obviously demonstrates that security may be achieved without unnecessary additional programming complexity.

Pliant (De Mendez, 2000) may thus be seen as a triad: (1) the *Pliant programming language and low level libraries*, (2) an *Internet applications suite*, and (3) the *FullPliant operating systems*. The Pliant programming language is human-oriented (i.e., its syntax is trivial). It is strongly typed, and its expressiveness allows the user to program in a high abstraction level. The language is also reflexive, allowing the user to change the way Pliant parses and compiles expressions. In other words, users have a high degree of freedom to redefine the Pliant language itself, should they dislike a particular feature of the language or want to extend it. The Pliant compiler is dynamic and efficient, producing code, on-the-fly, as efficient as the best C compilers, such as the C Compiler (GNU, 2003). The *Pliant Internet applications suite* consists of a set of servers (HTTP, FTP, SMTP, POP3, and DNS), a database engine, data encryption tools (RSA, RC4, Ciphersaber, and others), and a handful of Web application tools, such as the HTTP server configuration tool, an online forum, Web mail, printer configuration tool, to name a few. In the core of the suite is the Pliant HTTP server. This server is in charge of hosting and dynamically translating Pliant Web pages, written in Pliant, into HTML (HyperText

Markup Language), thus enabling mainstream browsers as the default Pliant applications user interface. The *FullPliant operating system* to be used by advanced users has two main goals: facilitate the system administration of a set of computers, and make it easy to automate repetitive tasks.

General Features of the Pliant Language

Standard Pliant applications are browser-based, i.e., the programmer can host his/her Pliant program in the Pliant HTTP server, and then interact with it using a Web browser. Browser-based Pliant programs have file extension *.page*, and are written using the *Pliant .page format*, an alternative to XML. The way the HTTP server works is as following: it keeps listening to requests from clients (i.e., other browsers on the Web). Once it gets a request for a page, it translates the Pliant program into an HTML and JavaScript mix and sends it to the client (browser). Hence, for the client, there is no difference. It is a simple plain HTML/JavaScript. In order to illustrate the simplicity and power of the Pliant language and HTTP server, let's present a simple example of Web application written in Pliant and the actual simplified version of its generated HTML code. Pliant has the capability of generating (and caching) online graphics when server-side font rendering has been requested. The client will see this application as shown in the figure at the right.

A Pliant program

```
title "A Pliant page"  
text "It is very easy to "  
eol  
text "write a Pliant page."
```

The translated HTML code

```
<html>  
<head>  
<title> A Pliant page</title>  
</head>  
<body><h1>  
<center>A Pliant page</center></h1><br>  
It is very easy to <br>  
write a Pliant page.  
</form></body></html>
```

The Web page



The command *title* produces a page title. The command *text*, whose argument is a string, outputs a text. Pliant provides a plethora of commands for writing Web pages. The interested reader should check the Pliant Documentation Initiative site (De Mendez, 2003) for the details. If the programmer wishes to insert an end of line, he/she must use of the command *eol*. This example shows that it is very easy to write a simple Web page in Pliant.

The next example illustrates how to include a computation in the page. The programmer writes a page that will show the result of the computation. This page is put under the shadow of the command *button*, i.e., indented with respect to this command.

```

title "Euro to Dollar"
var Float euro := 1
input "Amount: " euro
button "Press me"
  title "The answer is..."
  text (string euro*1.2)

```

The argument of the command can be a calculation. The function call *(string euro*1.2)*, for instance, transforms the numerical result of the expression *euro*1.2* into a string. The user of this page should type a number in the field and press the button, as shown in the figure below (left). A new page will show the result of converting the input value from Euro to Dollar, as shown on the figure below (right).

Dynamic Web pages in Pliant



In a short paper like this one, the authors may have failed to convey how easy it is to program in Pliant. To write a Pliant application of average complexity, including numerical computations (matrices and such), database management and graphics is also quite easy, thanks to the tighter than usual integration with the HTTP server. The authors have tested the “human orientation” of Pliant, asking people who are not programmers to write Pliant pages with short computations. For instance, accountants were asked to write a page that could perform interest calculations or portfolio planning. After a one hour tutoring, most subjects could finish the task. The readers of this paper are invited to test this affirmation themselves; playing with Pliant for about an hour and seeing what one can do with the language.

Pliant as a Tool for consolidating E-learning Management Systems

E-learning can be defined as the use of network technology, namely the Internet, to design, deliver, select, administer and extend learning (Masie, 2001). In this exercise, two basic types of technological solutions can be used, referred to as synchronous model (such as chat/IRC, whiteboard, audio-video streaming, videoconference), and asynchronous model (such as hypertext publication (i.e. www), e-mail, mailing lists, newsgroups, bulletin boards and file download (ftp)). Existing e-learning management systems, such as WebCT, Lotus Learning Space, Blackboard, Centra, etc., incorporate both models and corresponding services, in different ways. However, due to certain limitations proper to system stability such as: troubleshooting, file format accommodations, web browsers, customization, etc. None of these platforms include support to help manage the dynamics of e-learning activities (IMS, 2003). In an attempt to overcome these disadvantages, the study of management issues in e-learning environments has become critical for the success of Internet based

educational services. As an example, it has been established that such study could increase the corporate e-learning market in Europe by 96% in the next four years, and speed up changes in the education and training systems (E-Learning, 2000), (McGovern, 2000). A recent research work on the management of e-learning environments (Ramos, 2001) has shown the means and effectiveness of using the workflow concepts, techniques and tools to help manage e-learning.

Under this concept (E-Workflow, 2003), the e-learning process is considered as a two level inter-related process: the e-learning environment and the e-learning activities. The e-learning environment is breakdown into four different phases, each with its own set of tasks. These phases are described as following:

- (1) Conceptual phase, which refers to course subject, target audience, contents, budget organization, etc.
- (2) Planning phase, referring to details for the establishment and preparation of a specific instance of an e-learning action.
- (3) Execution phase: period of time during which the students are active in the learning process.
- (4) Procedural evaluation phase, referring to the analysis of how the e-learning fulfilled its aims. Additionally, the workflow e-learning environment model incorporates features such as improved efficiency, better process control (i.e. standardization of working methods), improved users service (i.e. greater predictability in levels of response to users), flexibility (i.e. ease of re-design in line with changing needs),
- (5) Business process improvement, i.e., streamlining and simplification of processes.

On the other hand, the e-learning activities are concerned with the monitoring of actions and interactions among the above described phases. They are controlled by means of "management elements" (the so-called "learning objects" (E-Learning, 2000)). Under the workflow model, the efficiency of an e-learning management system is measured by its capability in reusing the learning objects. To this end, Workflow software tools, such as Lotus Notes, Microsoft Exchange and others, were considered in the implementation of a preliminary framework (Ramos, 2001), along with XML (eXtensible Markup Language) as the language used for the description of learning object (World Wide Web, 2003).

Any attempt to provide implementation techniques that could promote the deployment of re-usable contents (i.e. learning objects) for e-learning purpose, while enhancing the survivability and integration capabilities of its underlying software platform, is therefore highly desirable for both the providers of educational services and the e-learning research community. The Pliant language is an "excellent" candidate that can fulfill such requirements.

Pliant as an Enhancement for e-learning Management Systems Platforms

Any existing e-learning management system is driven by means of its underlined software platform (called programs). At a human level, programs appear as algorithms, that is, a list of tasks, each being expressed by a single word or by a full sentence with subordinate clauses referring to subtasks. At a high (symbolic) level, a program is stated as a list of expressions. Each expression is raw data characterized by its semantics. At a low (i.e. code) level, a program is a set of instructions, where each instruction is a function call with a set of parameters. At this level, no ambiguity should remain.

A programming language is a bridge between the human way of thinking of an algorithm and the computer way of coding a program. Prior languages, including those on which the existing e-learning management systems are built, focus either: (1) on the semantics, but fail to be efficient at code level, (2) or on efficiency, hence failing to be intuitive and easy to use by a human. To our knowledge, Pliant is the first language that makes the "single language" option a possible one by acting as a bridge between the two last levels. We assert that Pliant is the best one available, because it addresses this bridging goal at the highest level

of flexibility and the best level of efficiency.

This ability to write all code using a "single language" means better internal communication, time saving improvements, load sharing, shorter code. It also means reasonable scalability, adaptive user interface, easy switching from a closed software-like model to an open-software like model, more flexibility, customization, development power, strong design and high quality program, low cost, dynamic highly reflexive compiler, less hardware limitations, adaptive hardware, etc. The list of goods is long. In order words, Pliant tries to bring as much expression power as possible, without impacting low level code performance. These capabilities allow Pliant to be seen as a kit that greatly simplify the distribution of software, hence can be considered as a "suitable" candidate for the development and deployment of e-learning management systems software platforms. In addition to the above capabilities, Pliant can also offer e-learning potentials, such as Internet-based learning and didactical requirements.

Findings

As a developed Web technology, Pliant is an Internet suite [11] that contains everything needed to start an Internet site, including a database engine, a forum, a graphical toolkit, dynamic pages and mail support. Therefore, it provides a suitable multimedia support to teachers and students, just as other proven e-learning



management systems, but with the added flexibility and adaptability of the underlying software and hardware platforms as pointed out previously. Pliant provides the choice to select elements according to the needs of teachers and students, and independently of their program of study. These elements can be divided into three groups, classified as: (1) elements for exploration and data management, (2) elements for teaching and communications, (3) and elements for user's management.

Elements for Exploration and Data Management

Pliant allows for a selection of various types of advanced web browsers: Netscape, Mozilla, Galeon, IE, Konqueror, Opera, and is open to other better ones available. The choice of a browser depends on the security, portability and computer power requirements. It also provides various types of servers suitable for "Internetworking", such as HTTP, FTP, DNS, SMTP, POP3, Web-mail, backup system, files browser, database engine, etc. These elements constitute a set a rich environment provided by the *FullPliant* operating systems for the data management and user interface operations.

Elements for Teaching and Communications

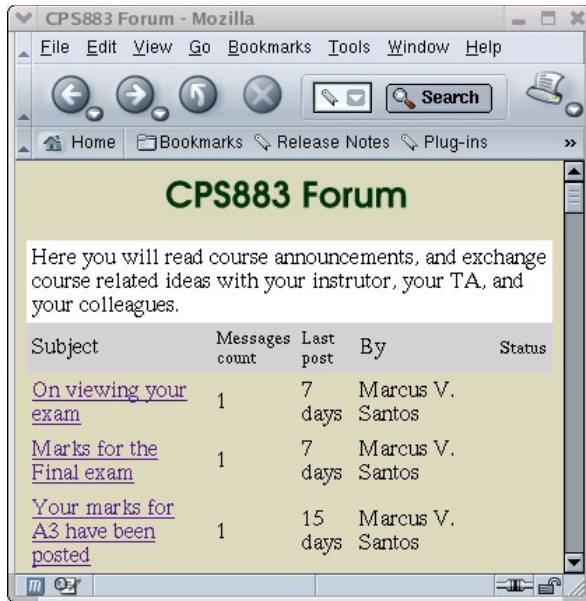
Pliant can be used as a tool for enhancing Course Management Systems. It provides a flexible and adaptive software support for a variety of learning processes such as the distribution of documents and communications through the Internet. The Pliant's online forum application provides asynchronous communication between instructors and students. The simple structure of the language allows one to easily create course Web sites on which he/she can post course notes for anytime access by students. With some training in Pliant programming, novices can quickly move on to the development of their own handy, tailor-made teaching tools, such as course assessments, online tests, and online grade book. To illustrate the point, next we show a typical simple course Web page written in Pliant, and its result.

```
title "A Course Web page"
table columns 1 border 0
cell
  link "Course Management Form" "cmf.pdf"
cell
  link "Announcements" "announ.html"
cell
  link "Forum" "forum.html"
cell
  note "Course Work"
  title "Course work"
  list
    item [Marks]
    item [...]
```

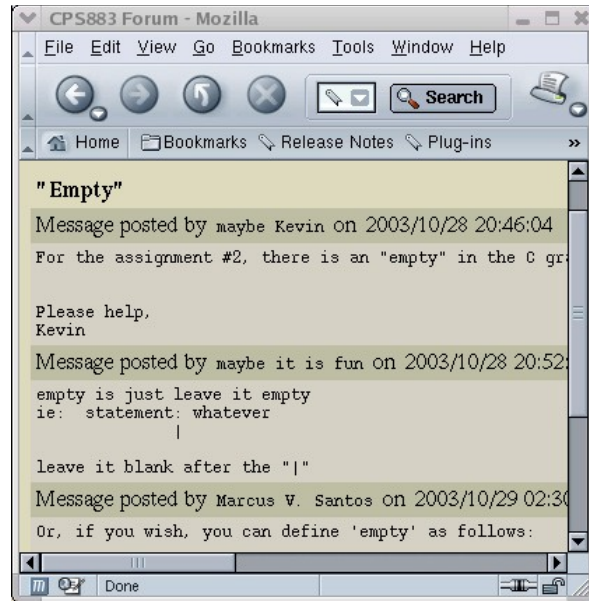
The example shows the options provided: (1) *Access to course documents*, in this case exemplified by a links to the course management form and the course work, but can be a list of documents, such as readings, lecture notes, schedules, syllabus, course management form, organization of course projects, priorities and details, etc. It also supports the import and export capabilities by means of inserted Web site links, allowing the instructors to gain access to complete set of teaching tools provided by academic publishers, or to create a package of the course content than can later be imported into another course, (2) *Course announcements*: a key place to put daily, weekly, or monthly time-sensitive course information such as deadline changes, clarifications, remainder of upcoming class chats, schedules, important events and dates, etc., (3) *Forum*: a web portal that behaves as a virtual classroom and lightweight chat. It enables users to participate in an on-line collaboration with students and instructors. As a discussion board, messages are posted to the board and every permitted user is able to read the messages and reply to them. Like a bulletin board, one copy of the message exists, and only the course designer has the right to delete the messages. A forum is a tool that fosters communication and collaboration as a means to enhance course material. Several forums can be created simultaneously, providing for a frame for team working. Each forum is assigned a thread (i.e. a discussion session) so that all replies to a given message is contained within the same thread. Within a forum, a messaging program is implemented, that allows to send email messages to the users who are members of the forum, and to keep track of those messages. As a collaboration tool, a forum allows it users to enter a real-time discussion with instructors, students, and colleagues, to access the Web, and to engage in question and answer sessions. The option of considering grouping student lists into several small groups can also be applied to keep the conversation manageable due to the synchronous nature of the discussion forum panel. It is important to point out that collaboration sessions throughout forums are recorded by means of subjects and messages. The leader of the session (i.e. the actual

course designer) must start the recorder to create an archive. Some of the above described features are highlighted in the figure below.

A forum front page



One message thread in the forum



Elements for Users Management

In an e-learning prospective, the user management capabilities of Pliant is mostly reflected on the ways Pliant enables the Instructor to manage the users in their course site. This involves the following kind of setting privileges: enrolling users in the course (which means that the user must already have an account), removing users from the course, creating groups of users within a course with the right to modify groups. Here, the instructor has the option of giving the group access to their own private Discussion Board, Virtual Classroom, Group File Exchange or Group Email.

Conclusions and Future Work

We have described Pliant as a "standalone" and "Web-based" language that encapsulates both the "human" and "computer" levels of thinking and coding programs. This unique privilege makes it an exceptional language, compared to any other existing one. It also demonstrates a higher level of flexibility, adaptability, and integration, providing for higher software development capabilities and enhancements. Also thanks to Pliant's HTTP server power (including server side rendering), any mainstream Web browser should always be enough to access Pliant documents, and by extension, Pliant e-learning materials. These qualities can be exploited to improve the current-state-of-the-art e-Learning development tools, while meeting educational expectations, economical, time constraints, and human resources limitations. The main advantages of Pliant over other integrated software solutions are: high transferability, flexibility, and maintainability. Components under free license agreements are freely distributed over the Pliant Web site. A marketing of the Pliant product results with commercial licenses is planned for the future.

The Pliant browser is expected to be released next year. It is expected to overcome the poor interactive capabilities of current today's complex HTTP + HTML + DOM + Javascript + SVG protocols set. It will bring state of the art user interface through a fairly simple protocol. Also, it will keep the run-on-everything capability by adopting VNC protocol, and be the bed for applications, thus easing the production of e-learning material and every day desktop productivity.

References

De Mendez, M., & De Mendez, P. O., & Santos, M. V., & Tonneau, H. (2003). Pliant Documentation Initiative, <http://pliant.cx/pliantdocs/babel/universal/>

De Mendez, P. O. (1999). Pliant: Expressive power plus efficiency". *Presentation at the ALCOM-IT Workshop and Review Meeting (Final)*, Barcelona, June 1999.

Hochheiser, H., & Shneiderman, B. Universal usability statements: Marking the trail for all users", *ACM interactions* 8, 2 (March-April 2001), pp. 16-18. Available online at <http://www.acm.org/pubs/citations/journals/interactions/2001-8-2/p16-hochheiser/>

GCC Home Page (2003). GNU C Compiler, *Free Software Foundation (FSF)*. Available online at <http://gcc.gnu.org/>

Masie, E. (2001). A Presentation on e-Learning Europe 2001-Dublin, Ireland. http://www.techlearn.com/dublin/followup2001/SLIDES_files/frame.htm

IMS Global Learning Consortium (2003). <http://www.imsproject.org>

E-learning (2000). Design Tomorrow's Education, *A European Commission Initiative to an eEurope*, <http://europa.eu.int/comm/education/e-learning/index.html>

McGovern, S., & Sarraf, B. (2000). Corporate E-learning, *Market Forecast and Analysis 2000, IDC Forecast, October 2000*, <http://www.itresearch.com/alfatst4.nsf/unitabsx/WTT16G?openDocument&q=tt16g++++>

Ramos, F., & Moreira, A., & Santos, A. Towards a Reference Framework for E-learning Management, *Proc. of DLA2001-Distance Learning Administration Conference*, University of West Georgia, GA, USA, 6-8 June 2001.

E-Workflow (2003). The workflow Portal, <http://www.e-workflow.org/>

World Wide Web Consortium -W3C (2003). XML Specifications, <http://www.w3.org/XML>

De Mendez, P. O., Santos, M. V. , & Tonneau H.. The Pliant Web site (2000), <http://fullpliant.org>

