

# The T++ Approach to Web Application Development

Antonio Soares de Azevedo Terceiro , Christina von Flach Chavez

Departamento de Ciência da Computação,  
Instituto de Matemática – Universidade Federal da Bahia  
Campus de Ondina Av. Ademar de Barros, Ondina – 40170-110 Salvador, BA  
{terceiro,flach}@im.ufba.br

**Abstract.** *C++ is a general purpose programming language widely used for developing large and efficient applications. Despite its usefulness in the development of a broad range of applications, there is no established platform for developing web applications in C++. This paper describes T++, a tool that supports web programming in C++. T++ allows the developer to combine C++ language statements with static content (e.g. HTML) in order to produce dynamic content, assisting in the task of deploying new and existing C++ applications on the web.*

## 1. Introduction

As the demand for Web applications grows, so does the demand for tools that support Web application development. *Web application development* comprises the process of using programming tools to implement web sites with dynamically generated content. The most popular tools for developing web applications include (i) *programming languages* specifically designed for Web programming, such as *PHP* [2], and (ii) extensions to general purpose programming languages, such as *Servlets/JSP* [1]. General purpose programming language extensions provide an additional benefit: they can be used to amend conventional applications (that is, originally developed for desktop systems) for the web environment.

The *C++* programming language [7] has been used in the development of many types of applications, from simple tools to complex information systems, including scientific and industrial applications. However, up until now, there is no established platform for developing web applications in *C++*. The lack of such a tool prevents the development of new *C++* applications for the web environment; moreover, it hampers the deployment of a solid bulk of *C++* applications on the web.

This paper presents *T++*, an engine designed for running web application programs written in *C++*. *T++* does for *C++* what *Servlets/JSP* have done for Java: it allows the development of new web applications written in *C++*, using all *C++* resources as well as the adaption of conventional desktop applications for the web.

The rest of the paper is organized as follows. Section 2 presents an overview of *T++*, describing the *T++* programming language (Section 2.1) and its execution engine (Section 2.2). Section 3 depicts some conclusions, stating the contributions and future work.

## 2. The T++ Approach to Web Application Development

*T++* is a tool for web application development in *C++*. It is free software, licensed under the GNU General Public License [5]. *T++* comprises a *programming language* and an *execution engine*. Running *T++* requires a GNU/Linux system, an Apache web server, and the GNU Compiler Collection (gcc) with support to *C++*.

$T++$  applications can be regarded as web server dynamic extensions [6]: the  $T++$  engine is installed in a web server so that  $T++$  documents can be executed by the  $T++$  engine. The  $T++$  engine receives users' requests to a  $T++$  document, and generates responses from the  $T++$  document to the user. Users can access  $T++$  documents in the same manner they access other kinds of documents available on the web.

The steps for executing  $T++$  applications from the users' perspective are presented in Figure 1 and summarized below.

The user sends *HTTP* request for a  $T++$  document using a web browser(1) and the web server forwards the request to the  $T++$  engine (2). After that,  $T++$  maps the requested document to an object in memory (3) – see Section 2.2 for further details. If it is necessary, this object is (re)built, that is, the document is translated and deployed, and a new object is created (4). The mapped object is then used to process the user request (5) and to generate a response, often in *HTML* format (6). Finally, the generated response is sent back to the user (7).

Figure 2 and Figure 3 present a simple  $T++$  application. The program reads a data file containing  $(x, y)$  points and shows its content as a histogram. An excerpt of the source code - the one that generates the histogram chart – is presented in Figure 2. The generated response, as the user will see it, is shown in Figure 3.

## 2.1. The $T++$ programming language

A  $T++$  application consists of one or more  $T++$  documents. A  $T++$  document is a text file with static content – often *HTML* – mixed up with *C++* code. A  $T++$  document consists of one or more blocks, where a *block* is a piece of code surrounded by delimiters. The  $T++$  language provides several types of blocks, including:

- **Code execution blocks**, delimited by `<%` and `%>`. These blocks contain *C++* statements, to be executed when the embedding document is requested.
- **Output generation blocks**, delimited by `<%=` and `%>`. These blocks specify a *C++* expression, to be evaluated and written to output, replacing the original block.
- **Member definition blocks**, delimited by `<%!` and `%>`. These blocks contain the declaration of additional members – attributes, methods, etc. – to be placed within the class generated by the translation process.
- **Extern declaration blocks**, delimited by `<%#` and `%>`. These blocks contain *C++* declarations, often preprocessor directives (like `#include` or `#define`).
- **Comment blocks**, delimited by `<%--` and `--%>`. These blocks encapsulate source code comments. Analogous to other languages, their content is silently ignored.

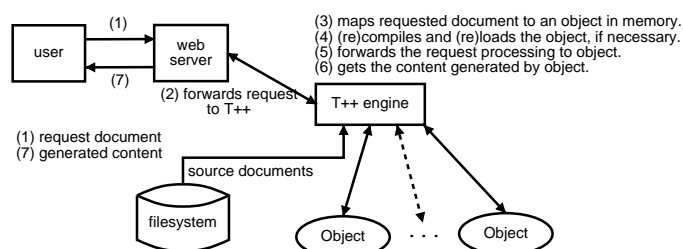


Figure 1: The processing of a request to a  $T++$  document

- **Static content blocks**, comprising everything else not listed above. These blocks are simply copied to output.

Figure 2 presents a sample *T++* source document. It consists of ordinary *HTML* static content (`<div>`, `<table>`) mixed up with *T++* code (one execution block, delimited by `<%` and `%>`, that defines a *C++* `for` statement).

Since any *C++* statement can be in a code execution block, *T++* can perform any operation that can be performed in a regular *C++* application, or in any other general purpose programming language.

*T++* documents must be translated into *C++* classes in order to be executed. As a result of the translation process, a *C++* class is generated with a method named `service()` defined in its interface. This method is incrementally built by reading blocks from the *T++* source document and generating the corresponding *C++* code, in the same order as the blocks were read. The exception for this translation procedure refers to member definition blocks and external declaration blocks whose content must be inserted at the beginning of the generated *C++* source file.

## 2.2. The *T++* execution engine

The *T++* execution engine is responsible for handling requests for *T++* documents and providing responses to the users. After a request for a *T++* document, the *T++* engine performs steps 3 to 6 (Figure 1). The execution engine maps the requested document (3) into an object in memory. This object implements a `service()` method, that will be invoked (5) to perform business logic operations, and to generate a response to be sent to the user (6). The translation of *T++* documents is transparent for the user and is automatically performed by the engine when a *T++* document is requested for the first time, or if the last version of the *T++* document available is more recent than the corresponding mapped object in memory. After translation, the generated *C++* class is compiled, a new object is created and (re)loaded into memory (4).

## 3. Conclusions

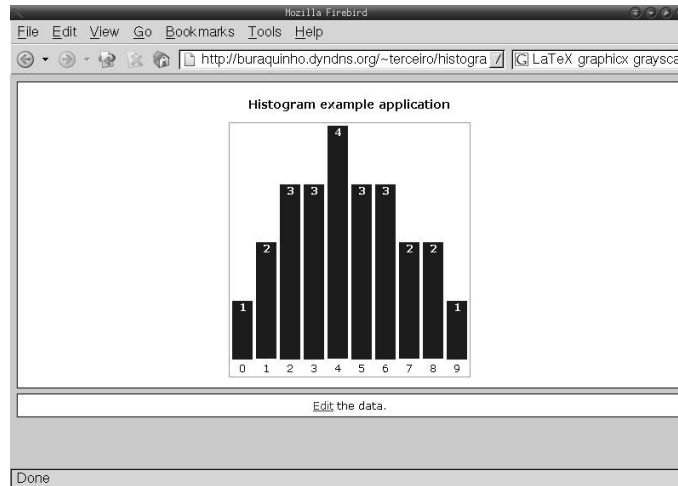
*T++* is a tool that supports web application development in *C++*. *T++* allows either the reuse of existing desktop applications in web environment, or the development of entirely

```

histogram.tpp (C:/webmidia2003/example) - GVH
File Edit Tools Syntax Buffers Window Plugin LaTeX Help
{ %>&nbsp;<%
%></di></td><%
--%>
<div class="chart">
<h3>Histogram example application</h3>
<table class="histogram" align="center">
<tr>
<%
for (int i = minX; i <= maxX; i++)
{
if (histogram[i] > 0)
{
%><td><div class="frequency" style="height: <%= (histogram[i] * 100)
/maxY %>;"><%= histogram[i] %></div></td><%
else
{
%><td><div class="nullPerquecy">0</div></td><%
}
}
}
</tr>
<tr class="xAxis">
<%
for (int i = minX; i <= maxX; i++)
{
%><td><%= i %></div></td><%
}
}
</tr>
</table>
</div>
43,11 89%

```

Figure 2: *T++* source code for the histogram example application



**Figure 3: The output of the histogram example application**

new web applications written in  $C++$ .  $T++$  is unique and different from other existing solutions and its implementation issues [4] were subject of interesting research.

Future work include the prevision of support for: (i) use of non-standard libraries, allowing the development of *front-ends* for existing applications, as well as separating new applications business logic from  $T++$  documents; (ii) code abstraction tags, a way to avoid explicit coding of common programming language constructs, such as control flow statements and output operations (this feature will avoid over-mixing  $C++$  code and static content by providing a template system for output generation); (iii) use of distributed components, allowing the development of distributed multi-tier web applications using  $T++$ .

We intend also to report experiences on migrating existing desktop applications to the web environment using  $T++$ .

Additional information about  $T++$  – such as, documentation, source code and others – can be found at [3].

## References

- [1] Javaserer Pages Technology website. <http://java.sun.com/products/jsp>.
- [2] PHP Hypertext Preprocessor website. <http://www.php.net>.
- [3]  $T++$  website. <http://tplusplus.sourceforge.net>.
- [4] Antonio Soares de Azevedo Terceiro and Christina von Flach G. Chavez. *T++: a Tool for Web Application Development in C++*. Accepted demonstration for OOPSLA'03, Anaheim, California, USA.
- [5] The Free Software Foundation. The GNU General Public License. <http://www.gnu.org/licenses/gpl.html>.
- [6] Sun Microsystems. The Java WebServices Tutorial. <http://java.sun.com/webservices/docs/1.1/tutorial/doc/>.
- [7] Bjarne Stroustrup. The  $C++$  programming language. <http://www.research.att.com/~bs/C++.html>.