

# Workshop, Commercializing Ada

Robert C. Leif

rleif@rleif.com www.newportinstruments.com

## Ada & Visual Studio.net

The first major topic discussed was the need for a commercial grade Ada compiler that was fully integrated into Microsoft® Visual Studio.net. I pointed out that Eiffel was already available in this format and that this was written up on the Microsoft website ([http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/pdc\\_eiffel.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/pdc_eiffel.asp)). Software that runs under Windows is the largest commercial market. Significant sales in the Windows' market require a commercial grade Ada compiler, which has passed the conformity assessment suite and the many other tests that compiler vendors use. The compiler would have to output Microsoft Common Language Specification compliant code and be integrated with Visual Studio. This would involve extending the Visual Studio editor, debugger, and possibly other tools to be Ada aware or integrating these tools to run under Visual Studio. The extension of GNAT to A# by Martin Carlisle has demonstrated feasibility. However, the transformation of A# into a commercial grade compiler and its integration with Visual Studio is far beyond the capabilities of an individual, albeit, very gifted researcher.

### XML\_IO

Since the present duality of operating system and web applications is evolving into a combination of the two, the dominant form of input and output will be XML. Since Ada has the advantage of having data-type semantics that are very similar to that of XML schema, it has been possible to create identical data-types in both languages. Text\_io should be extended to XML\_io. There are existing Ada packages that interface with XML. These include XML\_EZ\_Out by Marc A. Criley of McKae Technologies ([www.mckae.com](http://www.mckae.com)), Xml\_Automation by Mário Amado Alves ([www.softdevelcoop.org/software/xml\\_automaton/](http://www.softdevelcoop.org/software/xml_automaton/)). The number of software development tools for use with Ada would be expanded by developing technology that would permit the editing of Ada sources in an XML environment.

The creation of formatted tables with Ada is painful. XML has the advantage of separating the formatting from the code that generates the text. Possibly Dale Stanbrough's HTML packages (<http://goanna.cs.rmit.edu.au/~dale/software/>) could be integrated into the aforementioned Ada XML packages. A standard method needs to be developed to call Ada from web pages. Ada needs to have web page capabilities similar to JAVA & Eiffel.

Similarly if Ada vendors must produce products for other languages, they should participate in the very large XML market, where they will have the advantage of being able to reuse Ada technology. For instance there is a need for import (withing) diagrams to analyze the relationships between schemas. Ada technology could be applied to creating reliable efficient versions of XML's specialized languages, such as XSL, XSD, XForms, XUL (XML User-interface Language).

### Copyrights

Conventional software copyrights and distribution techniques do not include the sources. Although this is easy on manufacturer, it often because of a lack of peer review results in poor quality control and a lack of reliability and maintainability. Both GNU & Progressive software require that the sources be available. However, there are significant differences between the two approaches. GNU is monopolistic and Progressive (<http://www.softdevelcoop.org/>) is based on free market principles. I argued that Progressive Software is the next step beyond Free Software, since its licenses combine the best of open source and copyright law.

Stallman was a prophet when he demanded that the sources be made available. Progressive has retained the concept from conventional software that the developers be rewarded for their work with either directly with money or future royalties, or both.

The capability to inexpensively with low overhead develop commercial software may be very important to software engineers after their job has been outsourced. From the point of view of the customer, economic reward to developer is the best guarantee to the user for continued interest, maintenance, improvement, and support

Ada Core with GNAT Pro has adopted the tenets of progressive software.

“Your rights to use or adapt GNAT Pro components”

“You have complete freedom to include in your product any of the GNAT libraries or run-time components, as is or with your changes. You can do this without any obligation to expose or distribute your source code. Thus you can use GNAT Pro to develop proprietary or otherwise restricted software products.”

Ada Core is now employing the GNU General Public License (GPL) as motivation for developers to pay for GNAT Pro licenses.

“You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.”

The requirement for GPL for nonpaying users is a strong inducement to become a paying customer. It should also be noted that the individual users and small companies who can not afford a GNAT Pro license have no market influence.

### **Distributed Development**

It is hoped that the economic advantages of low-cost Progressive Software will result in the creation of a large market for reusable software components based on a new economy of software development. A royalty based remuneration system has the advantages of replacing large Frontload (Money upfront) costs by Backload (Royalties). With Ada, it should be possible for much of the code to be developed under royalty rates that are based on an algorithm that employs ASIS technology rather than traditional negotiation, which often requires expensive lawyers and accountants. Remuneration of the developers is in the interest of the customers because it increases the probability that the software will be maintained and supported.

Ada, because of the capacity to compile specifications and the requirement that the body is consistent with the specification, is an excellent language for distributed development. Because the developer-capitalists can work at home and most of their remuneration would be in the form of royalties, the upfront capital costs should be minimized. The availability of high-speed Internet already permits software distribution at a negligible cost. Thus the initial cost of distributed development should be much lower than present high overhead commercial development, which includes very significant costs for the use of buildings and other high-overhead costs.

The Software Developers Cooperative (SDC) is a group that is engaged in fostering Progressive Software development. The SDC aims at combining the best of open source with the maintenance of intellectual property rights. The SDC website (<http://www.softdevelcoop.org/>) includes selected and drafted licenses that realize the SDC principles. It also includes software including Ada. The SDC has an open mailing list.