Piper: Peer-to-Peer Distribution of Scientific Programs and Data-Flow

J.W. Bizzaro*, Jarl van Katwijk, Jean-Marc Valin, Brad Chapman, Dominic Letourneau, Gary Van Domselaar and Deanne Taylor

* Corresponding author:
Bioinformatics.org
c/o Department of Chemistry
University of Massachusetts Lowell
Lowell, Mass 01854

jeff@bioinformatics.org

Abstract

Many scientists have considered command-line interface (CLI) programs to be simpler to develop than their graphical user interface (GUI) counterparts. Combined with the power and flexibility of the UNIX paradigm (make large and complex tools by linking small and simple ones), development of this type has remained popular for decades, resulting in a plethora of small, often not directly compatible, programs that are not simpler to *use*. The authors have developed Piper to resolve the usability problem without sacrificing power and flexibility. Additionally, Piper extends the UNIX paradigm to peer-to-peer (P2P) Internet-distributed computing. With Piper, programs, files, widgets, and so forth, can be Internet-distributed components, represented in a GUI as flow chart nodes. The user can then create networks by connecting these nodes with lines depicting links for data-flow, procedural steps, relationships, and so on. Networks can be used to wrap existing programs so that they can be connected and executed in an intuitive manner, plus they can be compiled into nodes, saved, and shared with other users. In addition, individual programs can be located on high-performance or dedicated computers for compute-intensive/expensive calculations. Piper is designed as a modular system using CORBA connectivity protocols. It uses XML to describe a network, and it uses the model-view-controller (MVC) paradigm. This design allows different user interfaces to simultaneously control the system, providing for such possibilities as voice control. Piper is developed using Python and C++. For more information, visit http://bioinformatics.org/piper/.