Mawl: Integrated Web and Telephone Service Creation

David Atkins, Thomas Ball* Thomas Baran, Michael Benedikt,
Kenneth Cox, David Ladd†, Peter Mataga, Carlos Puchol,
J. Christopher Ramming‡, Kenneth Rehor, Curtis Tuckey

Bell Laboratories

`mawlers@research.bell-labs.com`
`http://www-spr.research.bell-labs.com/~mawl/`

March 4, 1997

Abstract

Mawl is a language and compiler for programming form-based services in a device-independent manner. PML is a markup language and middleware for controlling and programming various interactive voice response (IVR) platforms using standard web infrastructure. The combination of Mawl and PML allows the creation of interactive services that users can access via a web browser or telephone. The ability to create such services in a single environment appears to be unique.

The Mawl language separates the specification of service logic from the specification of the user interface to be presented on a device. As a result, one can easily code a service that is accessible via a web browser and with minor modifications to only the user interface specification, make the service accessible via an IVR platform. Mawl draws on the principles of application language engineering to facilitate not just service creation but the entire software development life cycle, improving a service provider's ability to develop, monitor, analyze, administer, and modify form-based services.

The Phone Markup Language (PML) is a language and an architecture that enables telephone access to web services and the easy creation of IVR services using the web paradigm. The PML language is a dialect of HTML specialized to describe content for interpretation over a telephone. HTML or PML documents are served over a telephone by standard IVR platforms (or audio processing nodes); as in the hypertext model, the documents themselves may reside on any web server in the network, or may be dynamically generated. The PML middleware takes care of the tasks of fetching documents from the Internet and instructing audio processing nodes to "play" them. As a result, to program an IVR service, a