PeriProducer to VXML Conversion Utility

Dramatically cut IVR application development costs and time to market.

Are you looking for an easy, sensible way to convert you PeriProducer applications to VoiceXML 2.0?

M&C Associates now offers an automated coversion service, which can move you from PeriProducer to VoiceXML, saving you up to 60% of the time it would take to start from scratch. Continue to leverage your investment in your IVR applications, and enjoy the benefits VXML offers.

Here's how it works:

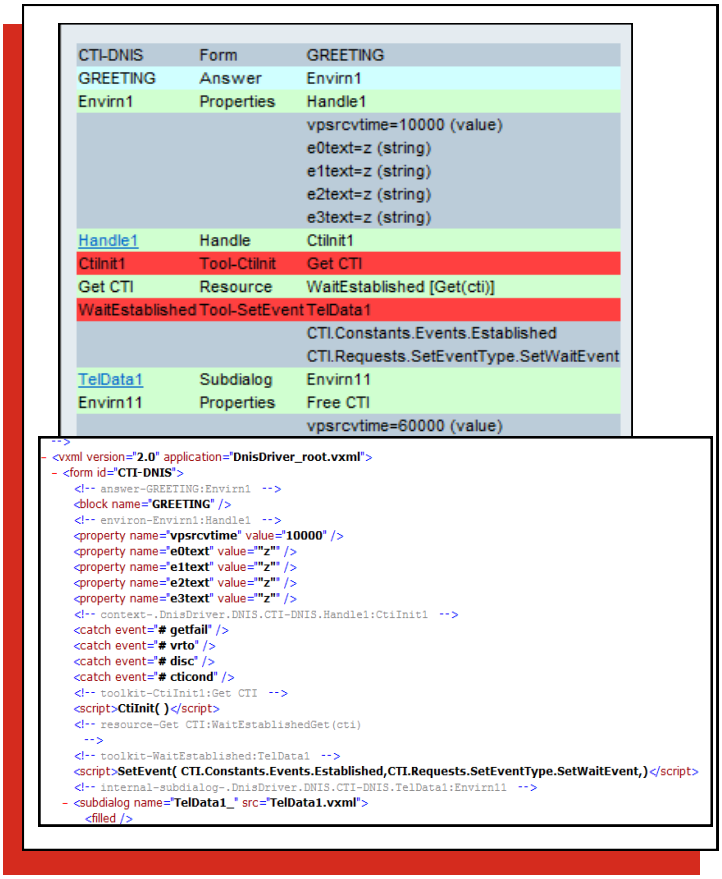
Original PeriProducer source code is stored as an ASCII file in a format that is equivalent to an XML document. In order to convert it, a series of automated analyses decompose the PeriPro source code into its components.

After decomposition, a multi-phase analysis abstracts component details and matches them against a set of different templates. The templates can be refined and expanded, so that the conversion process can increase its code coverage over time. A template match results in an equivalent VoiceXML document, with the details filled in from the PeriPro application.

One challenge to PeriPro conversion comes from the two part approach that VoiceXML takes to presentation and logic. VoiceXML divides an application into client-side presentation and server-side logic. Although PeriPro applications tend to have more client-side logic, they actual also divide into IVR and host-access components. Using the approach of VoiceXML wrappers, the amount of extra server logic can be minimized. By viewing the web server as equivalent to the "host" in PeriPro terms, an ap-

Why convert your applications to VXML?

- VXML provides unparalleled efficiency. It reduces system resource dependencies by up to 90% and improves calls per second (CPS) performance by 7X over traditional development tools.
- VXML is optimized for high density applications. It provides feature rich and high density media processing with call control signalling, IP and PSTN connectivity
- Offers ultimate, per channel value—lower TCO



CTI-DNIS	Form	GREETING
GREETING	Answer	Envirn1
Envirn1	Properties	Handle1
		vpsrcvtime=10000 (value)
		e0text=z (string)
		e1text=z (string)
		e2text=z (string)
		e3text=z (string)
Handle1	Handle	CtiInit1
CtiInit1	Tool-CtiInit	Get CTI
Get CTI	Resource	WaitEstablished [Get(cti)]
WaitEstablished	Tool-SetEvent	TelData1
		CTI.Constants.Events.Established
		CTI.Requests.SetEventType.SetWaitEvent
TelData1	Subdialog	Envirn11
Envirn11	Properties	Free CTI
		vpsrcvtime=60000 (value)

```
-->
<vxml version="2.0" application="DnisDriver_root.vxml">
 <form id="CTI-DNIS">
    <!-- answer-GREETING:Envirn1  -->
    <block name="GREETING" />
    <!-- environ-Envirn1:Handle1  -->
    <property name="vpsrcvtime" value="10000" />
    <property name="e0text" value="z" />
    <property name="e1text" value="z" />
    <property name="e2text" value="z" />
    <property name="e3text" value="z" />
    <!-- context-.DnisDriver.DNIS.CTI-DNIS.Handle1:CtiInit1  -->
    <catch event="# getfail" />
    <catch event="# vrto" />
    <catch event="# disc" />
    <catch event="# cticond" />
    <!-- toolkit-CtiInit1:Get CTI  -->
    <script>CtiInit( )</script>
    <!-- resource-Get CTI:WaitEstablishedGet(cti)
     -->
    <!-- toolkit-WaitEstablished:TelData1  -->
    <script>SetEvent( CTI.Constants.Events.Established,CTI.Requests.SetEventType.SetWaitEvent,)</script>
    <!-- internal-subdialog-.DnisDriver.DNIS.CTI-DNIS.TelData1:Envirn11  -->
  <subdialog name="TelData1_" src="TelData1.vxml">
     <filled />
```

plication goes to the host when it needs data. If the data comes back in a VoiceXML wrapper, then full VoiceXML document generation with accompanying server logic rarely needs to be created. This limits full document generation to those cases where VoiceXML has no allowance for client-side computation of a value, such as the attributes of a <property> element.

In general, the bulk of any PeriPro program should be programmatically decomposable into client and server components, with the client components convertible to VoiceXML in a semi-automated fashion. The Invores tool performs this decomposition to note the areas where server logic is needed. This provides lines along which to split the application. Then the client-side code is converted to VoiceXML, with a number of annotations that must be examined in more detail by a developer.

As an example this approach, we have taken the NumDemo application (selected solely because it is familiar to every PeriPro developer and available on every PeriPro installation), and run it through the analyzers to provide hints of the conversion process. The the results can be accessed by selecting the **Available** menu item. The displayed table lists the available converted applications, and the links provide access to the high level summaries, and to the conversion tracker.