



TENALEA IVRS – VoiceXML interface high-level design

Introduction

TENALEA IVRS is an extension to the basic TENALEA Randomisation system, to accommodate IVR (interactive voice response) interactions with users wishing to perform registration, randomization and other interactions in the TENALEA system with a standard telephone. A comprehensive design document is provided as part of the product lifecycle documentation. This document will be a short explanation of design decisions that aren't obvious from reading the code.

TENALEA IVRS is based on two important fundamentals: VoiceXML 2.0 and Envoy. TENALEA IVRS is based on VoiceXML 2.0, the specification of which is available from W3C (<http://www.w3c.org>). VoiceXML is an XML dialect for the design of IVR dialogs. Envoy is a telephony software platform that supports VoiceXML 2.0 and provides the technical basis of TENALEA IVRS.

Basic design

TIVR is based on several ASPX files housed in the alea_ws_dev application used to dynamically generate VoiceXML from the TENALEA database. The language of implementation is C#. The files are:

- IVR.aspx: The front end which the telephony platform invokes. IVR.aspx gets parameters from the querystring, and passes them to IVRService.aspx.
- IVRService.aspx: Contains all the functions to generate the VoiceXML dialogs.
- IVRRoot.aspx: To keep state between dialogs, VoiceXML requires the definition of a root dialog. This is IVRRoot.aspx. It also contains the definition in VoiceXML of several variables that have application-level scope.

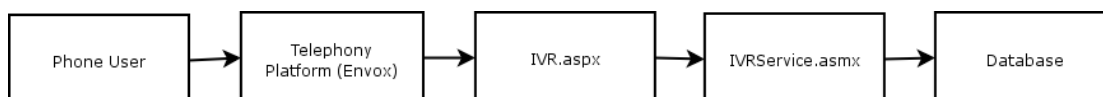


Figure 1. Chain of command.

The ASPX application is completely stateless, all the state is kept in the VoiceXML application which is run on the telephony platform (Fig 1). All communication from VoiceXML to the ASPX application happens via the querystring.



The VoiceXML application is divided into several dialogs, each of which requires an http roundtrip between the telephony platform and the ASPX application to obtain. Each dialog corresponds to a “mode”, and each mode has a name. Calling IVR.aspx

with the querystring parameter “mode=MODENAME” causes the fetching of the dialog corresponding to the mode MODENAME.

Navigation between dialogs happens depending on the input from the user. Figure 2 shows the different modes and the navigation structure between them.

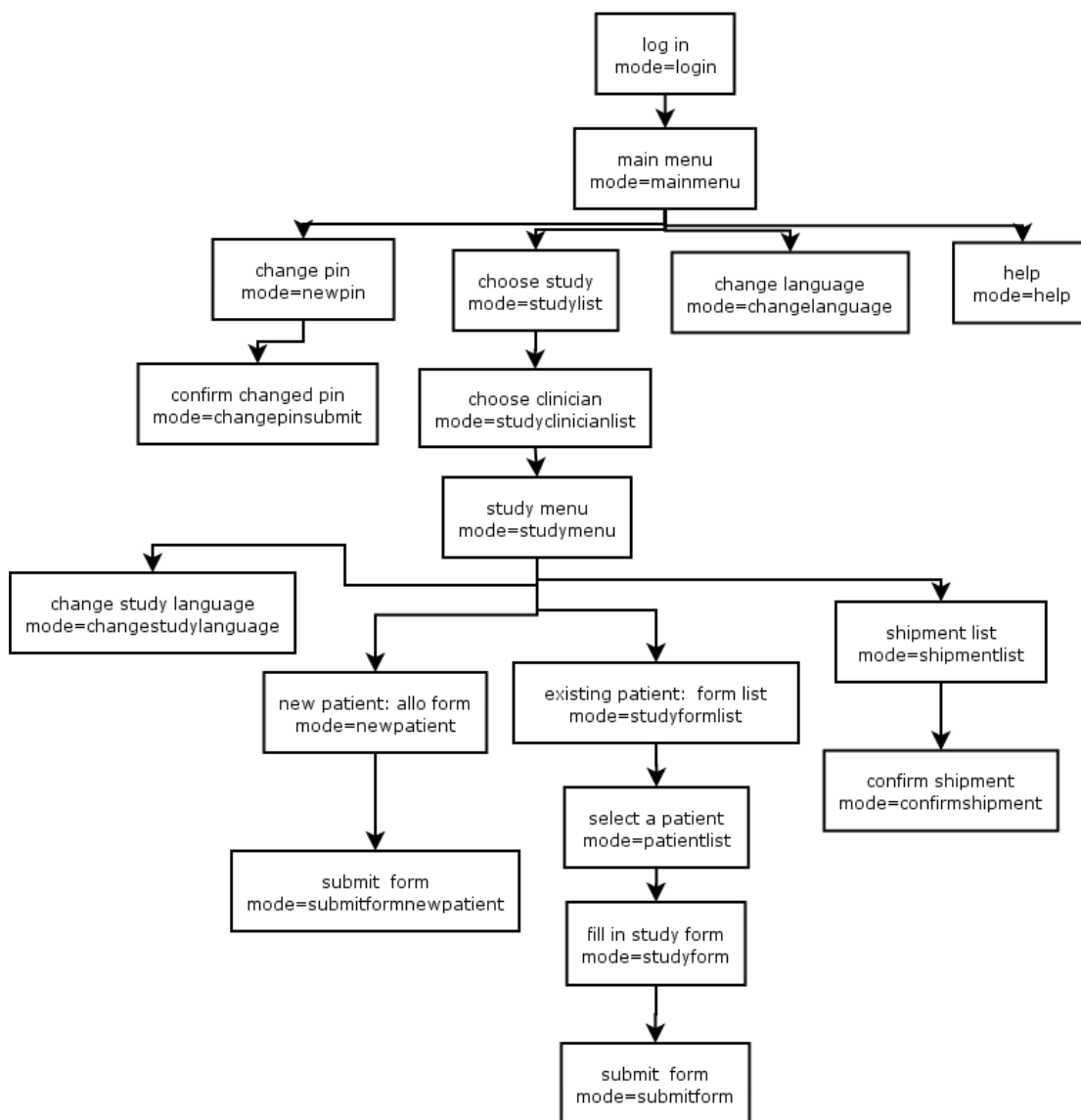


Figure 2: Dialog flow. The mode strings are given in the second line of every box.



VoiceXML Application-Level variables

IVRRoot.aspx defines the variables which represent the state of the application. The variables gradually get filled as the user advances through the application. The variables are.

- account_number: the ivr account of the logged in user
- pin_number: the pin number of the user
- language: the IANA ID of the language the user chose to interact with in the log in menu
- study_language: because the user language may not be available for a particular study, the language the user uses in a randomization can be different from his application level language. Study_language contains the language for study-level interaction. At the start of session, study_language is set to be the same as "language", and is only changed when a user chooses a study that doesn't support his language.
- study_id: ID of the study that user wishes to randomize for
- clinician_id: ID of the clinician that user wishes to randomize on behalf of.
- patient_id: ID of the existing patient that user wishes to fill in a form for
- form_id: study form the user wishes to fill
- shipment_id: for double blind studies, contains the shipment id that the user wishes to confirm'
- mode: contains the mode string of the current dialog (see above)

Each time a dialog is required, the voiceXML sets the mode variable to the required dialog mode, and sends a request to IVR.aspx, sending all the state variables in the process.

Example, to get the "mainmenu" dialog:

```
<assign name="mode" expr="'mainmenu'" />

<submit next="IVR.aspx" method="get" namelist="mode account_number
pin_number language study_language study_id form_id patient_id
clinician_id shipment_id" />
```

Functions in IVRService.ASMX

Dialog methods

All the modes described in the previous sections have corresponding methods to generate their dialogs in IVRService.ASMX. The methods all return an XML Element containing the VoiceXML application.

The method names are built up in the following way:

GetVXML[mode string]().



For example to mode “patientlist” corresponds the method GetVXMLPatientList().

Most data transfer between the database and TIVR happens via stored procedures.

Utility methods

IVRService defines a few useful methods for expanding the system.

- **GenerateVXMLMenu:** A general purpose method that builds a VXML menu from arrays of choices. It is quite flexible, the menu can be made to enumerate the options or otherwise; it can assign the result of the choice to a variable, and it can go to a different mode depending on each choice.
- **GetIVRMessage:** All the strings used in the application come from the UIStrings table. This method get a particular IVR string from that table.
- **GetMetaDataTranslation:** Gets strings from the MetaDataTranslations table. Used to translate study data
- **GetVXMLError:** Generates an error dialog