



Moving Enterprise Applications into VoiceXML

May 2002



ViaFone Overview



ViaFone connects mobile employees to enterprise systems to improve overall business performance.



- **Enterprise Application Focus; not Consumer**
- **Customer-facing Employees: Field Sales and Service**
- **Vertical Industry Focus**
- **Companies Have Enterprise Systems Installed: CRM, SFA, ERP, Email/PIM**



- Existing Speech Application Framework being moved onto VoiceXML



- All VoiceXML generated dynamically



■ Differentiating Speech Platform Features

- Adaptive Matching, N-Best Filtering, Constraints
- Natural Language Support
- Centralized Dialog Flow Description
- Declarative

■ Existing Vertical Applications

- MobileSales
- MobilePharma
- MobileService
- MobileAssistant

Enterprise Applications



- **Very Difficult to move to Deployment**
 - Recognition Quality and Dialog Design
 - Demonstration of Business Value
 - Commitment to ongoing maintenance

- **Characteristics**
 - Often very large, dynamic databases
 - Perceived easiest applications often the hard recognition/dialog problems
 - Not a consumer app, recognition must work well for everyone
 - Recognition must work consistently well for all items
 - (Lack of) Acceptance of technology limitations by USERS

The Problems



■ Recognition Quality

- Not just error rate but types of errors
- Rejections, user interface flaws perceived as recognition problems
- Names and non standard categories

■ Dialog Complexity

- Deceptive, even simple apps when fully fleshed out are surprisingly complex
- Specification of behavior and behavior very difficult
- Going far enough to make it worth it.

■ Spoken Language

- Problems get worse from pilot to deployment
- User Feedback rarely part of the design cycle
- Getting representative samples of speakers and data

■ Deployment and Maintenance

- Scale up for Actual Deployment
- Ongoing maintenance and tuning of grammars and dictionaries

A limited definition of “Natural Language”



- phrase variations
 - "10 servers"
 - "10 n-class servers"
- multiple "orthogonal" slots
 - "save email and send"
 - `<action1 "save email"> <action2 "send">`
- multiple "combinatorial" slots
 - "update it to 90%"
 - "update win probability to 90%"
 - "what is the win probability?"
- Clause variations
 - "90% for the win probability"
- Underspecification
 - "update probability" + "90%"
 - "set it to 90%"
- semantic and syntactic ambiguity

Natural Language in VoiceXML



- **Filled mechanism is weak**
 - Complicates generation when slots are “combinatorial”
 - Similar to overgeneration problem in grammars

- **N-Best Filtering**
 - N-best must be parsed independantly
 - No mechanisms for choosing amongst competing parses

- **Interpretation of Utterances**
 - Requires Application State Information
 - Insufficient Semantics Model (including proposal in 2.0)

Our Approach



- **Kept the Centralized Dialog Flow Engine**
 - 1 - to – many relationship from states to pages
 - Pages kept very simple
 - We do our own parsing for natural language support

- **Templatized Voice XML**
 - We generate a very simple skeleton
 - XSL Processing step to customize for different browsers
 - Customization of standard behavior in pages

- **Centralized Grammar Skeleton**
 - Grammar Sharing Mechanisms in XML, make testing and tuning difficult
 - Maintain one grammar with dynamically generated rules
 - Split the grammars for VoiceXML browsers

ViaFone Solution: Adaptive Matching



■ Data Preprocessing

- Abbreviations, expansion of numbers
- Combine descriptions of different features (fields from database)
 - For example:
 - “10 n-class servers for hp”
 - Becomes:
 - “ten one zero n dash class servers for h p hewlett packard”

■ Grammar Generation

- Very simple grammar structure

■ Application Logic

- Likely matches for recognition
- “Correction” for misspellings, autopron mistakes
- Application constraints to determine most relevant and/or likely matches

Why it Works



■ **Statistical Argument**

- For most mis-spelled or mis-phoneticized word, there is a similar word (or shorter word) that is correctly represented
- Requires a certain size of data set
- Paradoxically works better when grammar is large

■ **Application Constraints**

- Subsetting for each representative
- Integration with scheduler
- Recent usage, usage of related companies

Recognition: Example 1

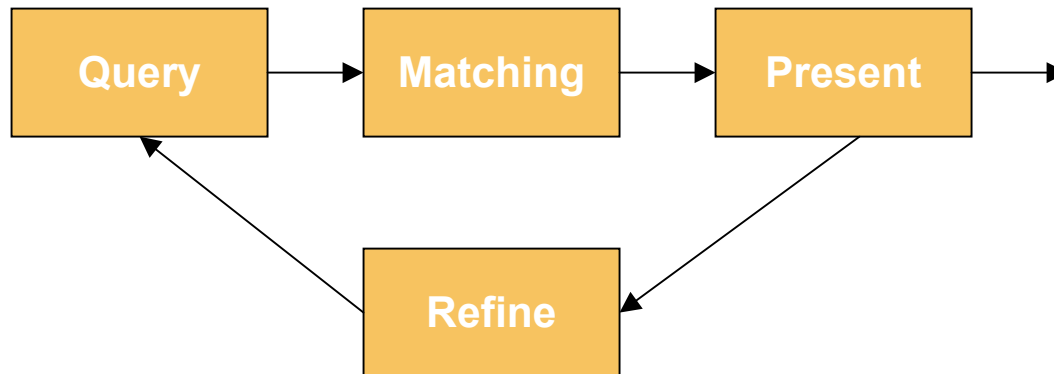


- **Selecting Opportunities from a Sales Database (e.g. Siebel)**
 - All opportunities have arbitrary names
 - Misspellings, Abbreviations, Duplicates, etc...
 - The key feature to the user may be contact, company, etc...
 - Sales rep can have 10 – 100 depending on industry, etc...
 - Many thousands of entries total
- **A type of “natural language” problem**
 - User’s don’t remember exact name, or they remember other features
 - Ambiguity in description
- **Conventional Grammar Solution**
 - Return a slot with id, phrasing variations (type A) hand-coded in grammar
 - 6 Man months effort (for a subset of 500 entries)
 - 60% Recognition Rate
 - Low coverage of phrasing variations

Solution



■ Dialog Structure



- 87% hit rate in first response
- 13,000 entries covering any variation in phrasing
- Searchable on multiple fields (name, contact, company)
- refinement of searches
- No rejections, matching errors more natural (for the most part)

Recognition: Example 1



■ Voice XML Template

```
<vxml version="1.0" application="http://...">
  <property name="universals" value="none"/>
  <form id="noname">
    <USE_NBEST/>
    <USE_DTMF params="..."/>
    <block>
      <prompt bargein="true"><audio src="http:..."/></prompt>
    </block>
    <field name="Result">
      <grammar src="http://..."/>
      <REC_SUBMIT next="http://...?SESSIONID=10964368" method="post" namelist="Result DTMFCount
DTMFStopTone nbestresult"/>
      <ERR_SUBMIT next="http://...?SESSIONID=10964368" method="post" namelist="Result"/>
    </field>
  </form>
</vxml>
```

Processed Template



```
<vxml version="1.0" application="http://...">
  <property name="universals" value="none"/>
  <form id="noname">
    <var expr="initial" name="nbestresult"/>
    <var expr="1" name="DTMFCount"/>
    <var expr="'pound'" name="DTMFStopTone"/>
    <block/>
    <field name="Result">
      <prompt bargein="true"><audio src="http://..."/></prompt>
      <property name="maxnbest" value="4"/>
      <property name="confidencelevel" value="0"/>
      <grammar src="">
        <nbest><![CDATA[
          for(var i = 0; i < lastresult$.length; i++){
            var inter = new String(lastresult$[i].interpretation);
            var len = inter.length;
            var idx = inter.indexOf('=', 0);
            inter = new String(inter.substring(idx+1, len-1));
            var conf = lastresult$[i].confidence;
            if(i == 0) {
              nbestresult = inter + '|' + conf;
            }else{
              nbestresult = nbestresult + ';' + inter + '|' + conf;
            }
          }
          return lastresult$;]]>
        </nbest>
      </grammar>
      <filled>
        <submit next="http://...?SESSIONID=10964368" method="post" namelist="Result DTMFCount DTMFStopTone nbestresult"/>
      </filled>
      <dtmf>+[ dtmf-1 dtmf-2 dtmf-3 dtmf-4 dtmf-5 dtmf-6 dtmf-7 dtmf-8 dtmf-9 dtmf-0 dtmf-star dtmf-pound]
      </dtmf>
      <catch event="noinput">
        <assign expr="#NoHear" name="Result"/>
        <submit next="http://...?SESSIONID=10964368" method="post" namelist="Result"/>
      </catch>
      <catch event="nomatch">
        <assign expr="#InputRejected" name="Result"/>
        <submit next="http://...?SESSIONID=10964368" method="post" namelist="Result"/>
      </catch>
    </field>
  </form>
</vxml>
```

Recognition: Example 2



■ User Authentication

- Large Database of Users
- Names have very non-standard phonetics
- PIN codes present technical, usability and security problems

■ Pure Recognition and Verification Problem

- Conventional grammar on names presents recognition problems
- Name + password helps, but users uncomfortable with speaking password

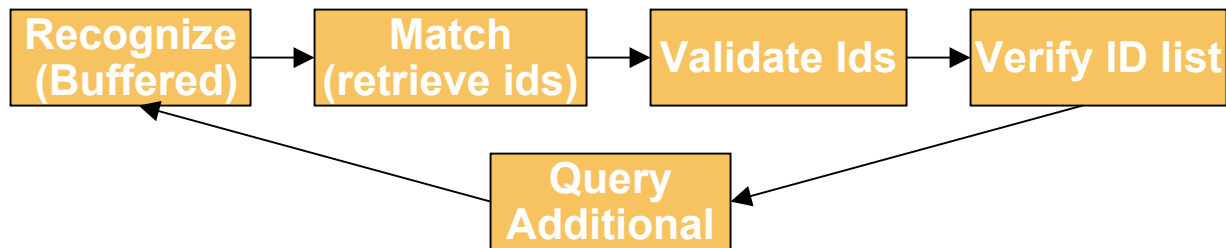
■ Adaptive Matching + Authentication

- Present cross constraints, verification is a filter on recognition
- Match list can be used to adjust confidence in verification results

Solution



■ Verification Process



■ Voice XML Template

```
<vxml version="1.0" application="http://...">
  <property name="universals" value="none"/>
  <form id="noname">
    <PLATFORM vendor="Nuance"/>
    <USE_NBEST/>
    <USE_VERIFY type="buffer"/>
    <block>
      <prompt bargein="true"><audio src="http://..."/></prompt>
    </block>
    <field name="Result">
      <grammar src="http://...">
        <REC_SUBMIT next="http://...?SESSIONID=10964368" method="post" namelist="Result DTMFCount
          DTMFStopTone nbestresult" />
        <ERR_SUBMIT next="http://...?SESSIONID=10964368" method="post" namelist="Result"/>
      </grammar>
    </field>
  </form>
</vxml>
```

■ Custom Speech Object for verification of multiple IDs

Deployment and Maintenance



- **Scale up**
- **Custom Grammar development**
- **Maintenance is ongoing and expensive**
- **Specialized expertise required**

Levels of Testing



- **Grammar Coverage**
- **Dialog Flow**
- **Correctness of Generated VoiceXML**
- **Application and Usability Testing**

Testing Support Tools



- **Static Analysis of Application Semantics**
 - Coverage of all recognition events
 - Listing all state transitions

- **Driver Application for Dialog Flow**
 - Validating browser behavior
 - Load Testing more representative of actual usage
 - Instrumenting the application

- **Http Request Validation**
 - Validation of application logic and flow independent of browser issues
 - Templates provide for easy checking and automation
 - Static analysis makes it possible to do full coverage
 - The real problem is the external specification of application behavior

Final Observations



Speech recognition is not an out of the box technology.

Look for the right balance between browser and server side presentation layer logic.

Consider the whole application life cycle when designing architecture. What can be done to make testing and maintenance easier and more automated.