Life Cycle and Development Methodologies

Eddie Kaul, Inference Communications
VP Professional Services
Current Challenges in Application development

• Need to demonstrate technology solutions in a cost effective manner.
  – No longer is a recorded audio demo sufficient for many customers.

• Demonstration systems are becoming an important part of the speech Tender process.
  – “Try before buy” is becoming commonplace
    • Run with focus groups
    • Run with stakeholders across multiple departments
More challenges..

• “Try before buy” becomes even more difficult when overlaid with current industry trends
  – Natural language
    • Less directed dialog
    • Moving to conversational dialog
  – Localization
    • Different languages
    • Different call flow
      e.g. Authentication: What’s a social security number?
So what option are there?

- Lets take a quick look at the current development methodologies to illustrate common problems..
  - and propose a new approach that mitigates them.
  1. Bespoke design - building from scratch
  2. “Prepackaged” – template driven, build once-sell-many
  3. Modularized – build from proven components
  4. Voice Management Platforms
  5. Grammatical Inference
1) Bespoke development

• Full design and development methodology
  – Standard software design methodologies
    • “Waterfall model”
      – Expensive
      – Tender has come and gone by the time your prototype is ready
  – Need to build it right the first time
    – Throw away prototypes are costly
  – Often necessary to ‘simplify’ an application in order to get it developed in the given time frame.
    – Directed dialog Vs Natural Language
2) Pre-packaged

• Pre-packaged / Template based / reuse
  – Can be viable. But assumes:
    • There is a template or pre-packaged solution to use.
    • You own it or have the right to use it.
    • It sufficiently localized for the task
    • The pre-packaged solution can be implemented in a cost effective and timely way.
      – Pre-packaged solutions are often bundled with the necessary VXML gateways, ASR platforms, infrastructure.

• Alternatively…
3) Modularized & 4) Voice Mgt. Platforms

- A popular approach is to use trusted components to build applications.
  - Dialog modules
    - Pre-defined components (e.g. credit card processing)
  - Or..
  - Drag-and-drop authoring methodologies
    - Voice Management Platforms
  - But can you get something you can talk to quick enough?
    - Call-flow should be ok
    - What about grammars?
    - What about natural language?
What about a new approach?

• Given the emerging trends:
  – Try before buy
  – A need for low cost prototyping
  – Natural language and conversational dialogs.
    • Emphasis on grammars
  – Addressing the concern of on-going maintenance costs

• What else could we use?
Grammatical Inference - the methodology

- A new approach where grammars and call flow are developed concurrently.
- Sketch the call flow then add the detail
- Use the sketch to define the underlying grammars
  - Example phrases train and build the grammar automatically.
- Focus of application development is “grammar centric” rather than call flow centric.
Why use it?

• Fast – very fast.
  – Full natural language prototyping in hours
  – Full VXML and Grammars
    • Context Free or Statistical Language Models (SLMs)

• The ‘sketch’ approach allows you to develop an entire application simultaneously.
  – Producing an interactive application as soon as possible.

• The more examples you provide the more sophisticated your application becomes.
  – Directed dialog utterances will develop directed dialog grammars.
  – Natural language utterances will develop natural language grammars.
How does it work?
Similar to a voice management platform

Use a palette of building blocks
..except call flow is roughly defined

Define overall call flow
..and grammar is specified via ‘fields’
Grammar is then trained via examples
Grammar development redefined

- Grammars can be defined as context free or SLM’s
  - Depending on the task at hand

- No need for:
  - Specialist grammar development expertise
  - Costly run-time environments
  - Thousands of training examples
    “a few utterances per dialog state”
But does it really work?

• A case study
  – Metlink
    • One of the largest public transport providers in Australia
    • Speech recognition implementation via a public tendering process
    • Gradual roll-out
      – “Try before buy”
        » Ticketing information system
        » Timetabling information
        » Ticket sales

• Inference asked to to develop a trial system.
• You have 3 days!
By the end of the first day..

- Platform independent demonstration system.
  - Timetable information
  - Service outages
    - Full natural language grammars
    - Recorded audio prompts
    - Context sensitive help
  - Used by multiple stakeholders throughout the company.
What did it sound like?

- Take a listen.
  
  metlink

- Tender won – game over.
Empowering Your customers and employees with speech technologies

SpeechTEKI 2006 Empower
The Voice Solutions Showcase

Thank You