‘Sorry I didn’t get that’ – an essay on automation, speech and stuff

Introduction

This essay is my[1] interpretation of what Bruce Balentine advocates in his 2 speech user interface books:

- It’s better to be a good machine than a bad human
- How to build a speech recognition system

I make no bones – I lifted almost all the material – however I referenced more than 1 book so it’s research based and not plagiarism!

Automation – the business case

Rule #1 – the real value of automation is cost saving – ‘fire not hire or redeploy’ – almost everything else is either spin or tenuous (i.e. ‘improved customer service/satisfaction’) – with the possible exception of improving access.

As such the primary motivation for automation is:

> *The automated delivery of service without sacrificing true customer satisfaction.*

…or put another way…

> *Lower costs without losing customers.*

It’s all about getting the balance right between cost and customer service.

The main ways through which automation can cut costs are by:

1. Enabling callers to self-serve.
2. Accurately routing callers to the appropriate agent/service.

NB: Since I downplayed its significance earlier - a quick word about ‘customer service/satisfaction’. Remember that people are calling with a goal in mind. <u>They will only ever be satisfied when they achieve that goal.</u>

> *Service = Satisfaction = ‘achieving your goal with minimal fuss’*

…hold that thought!

[1] Craig Scott

The bad news (Telephones)

Take the following mantra:

> *Speech is the most natural way for people to interact [with machines.]*

Well it's not true - generally visual interfaces (i.e. Windows) are better. To understand this statement we need to compare and contrast the different user interfaces:

- Telephone User Interface (a 'temporal' interface)
  - Narrowband – *not much information can be transferred.*
  - Serial – *information is transferred 1 piece at a time in strict order with no opportunity to skip/ignore.*
  - Occupies time – *occupies or wastes?*
  - Error prone – *recognition is a statistical process.*
  - No persistence of information - *significant effort required to simply remember the information – in addition to effort required to accomplish task.*

- Graphical User Interface (Windows – a 'spatial' interface)
  - Broadband – *huge amount of information can be transferred*
  - Parallel – *information can be selectively skipped/ignored (web adverts)*
  - Occupies space – *thus can be navigated up/down/left/right*
  - Accurate – *Once you've mastered the hand-eye coordination thing!*
  - Persistent - *no need to remember information – so can concentrate all effort on accomplishing the task.*

Basically TUI's are not good at transferring much information – and they're bloody hard work mentally (hence the phrase - 'temporal' interface)

More bad news (Speech Recognition)

Speech is an uncertain medium – it is extremely variable:

- Dialect/accent.
- Physical characteristics (vocal tract)
- Prosodic pattern (rhythm/stress level)

…and it changes as a result of the interaction itself (the more frustrated/stressed a caller becomes, the more their speech changes.)

Speech recognition is a statistical process – it will always suffer from false acceptance/rejection (i.e. regularly get it wrong)…

…and it isn't going to get much better – we reached the state of the art, as far as accuracy is concerned, a few years ago - there is not much more that can be extracted from the audio - and the next level (understanding) is not going to happen.

When people communicate they build up a mental model of the other person (empathy) – attempting to predict their behaviour. When they attempt this with a machine it can cause problems since the machine will rarely behave as expected – this makes the caller have to 'work harder' – thus creating frustration (which can lead to recognition failure)

You may ask why a human would attempt their 'empathy trick' with a machine – well we humans associate spoken language with intelligence[2] – and the machine is talking with us.

[2] Why else do we call stupid people dumb?

Final piece of bad news…

There are no standards (de-facto or otherwise) that define how a TUI should sound or behave. Every interface is different – with no consistency across application (often even with an application!)

We are still in the 'Wild West' as far as speech user interface design is concerned. This is largely because the speech industry has been constantly 'selling the future' (Natural Language, SayAnything, HumanTouch etc. etc.) implying that the problem with poor automation is down to the technology rather than the UI design and implementation.

So speech is a poor man/machine interface and bloody difficult to make work – which makes it all the more important, prior to embarking on automation, to ask the NAQ (Never Asked Questions)

- How important is it to solve this problem
- Why – what is wrong with the way it currently works. If something has value it should be possible to describe what that value is.

There is good news…

Now that we are fully grounded – understanding the business case for automation and the technical limitations of a telephone user interface - it is time to look more closely at where it is applicable and how best to approach the task – considering the following.

> *In the future, the importance of speech will diminish – but its value will increase.*

First piece of good news - some callers actually prefer automation. Frequent callers with simple repetitive tasks or those who want a degree of anonymity prefer it precisely because it is a machine! Automation tends to be faster, doesn't judge you or make you feel embarrassed or guilty and you are never 'wasting its time'. Examples might be:

- Getting test results (GU clinic!)
- Checking share prices - frequently.
- Paying bills – especially minimum payments (i.e. council tax)
- Cancelling appointments.

Next piece of good news – everyone has a telephone – and they carry it with them all the time – and to date the handset manufacturers and networks have not embraced the internet – partly through incompetence and partly through the limitation of the interface (small screen and no keyboard/mouse.) This is where the value of speech really shows – as a complementary interface (replacing keyboard/mouse?)

Another piece of good news – call centres are not going away anytime soon – and whilst there are rooms full of people answering calls, there are opportunities for cost saving through automation.

User Interface Design

Lesson #1

First thing to remember when designing a TUI, painful though it is:

> *Nobody is going to be 'delighted' by the user interface – they may tolerate it*

What 'delights' people is accomplishing their goal with minimal fuss. Once we accept this we can concentrate on getting the interface usable rather than 'likeable'.

> *The holy grail for a UI is acceptability.*

Remember, when people complain about automation – it isn't because they 'don't like the IVR' – it's because:

- They find it difficult to self-serve – thus a usability issue.
- They don't want to self-serve – there's no compelling reason to (which is actually a usability issue – it should be quicker and easier than queuing for an agent)

Concentrate most effort on error recovery (because errors are inevitable)

The biggest problems with speech recognition are:

- False Acceptance (FA) – *confidently misinterpreting what the caller said.*
- False Rejections (FR) – *not understanding a valid response.*

FA and FR are guaranteed to happen regularly in all speech applications – often without rhyme or reason (as far as the caller is concerned)

Once an error has occurred it is important to 'close it down' quickly. We must avoid a positive feedback loop – the effect of FA/FR is often a change in the callers voice (louder and slower) – which often results in another FA/FR – etc. etc. What makes things worse is that generally there is no way to tell that there has been a problem (we confidently got it wrong remember.) Even if we could identify that there has been a problem, there is no way to identify the reason (was it background noise, a genuine attempt, an unpredicted response or a secondary conversation.)

The solution:

- Rapid re-prompt (give the caller another go – quickly – no apologies)
- Give a single targeted hint after the first failure.
- Fail quickly to touch tone.
- Fail quickly to agent.

Persona's

Don't do it! By creating a persona, you are deliberately drawing attention to the UI itself and consequently drawing attention away from the task in hand. It would be like having a keyboard with flashing blue neon lights under the keys – look at me! look at me! Ooops – I forgot what was typing?

Also remember that by making the 'persona' anything but 'inert' will irritate as many people as it pleases[3]. Nobody was ever irritated by an 'inert' interface!

Branding and marketing

As with personas, in order for branding/marketing to work it must draw attention to itself – and hence away from the task in hand – so it's bad for usability. Spurious messages and adverts use up vital bandwidth and time – and unlike the web, cannot be skipped or ignored. What is the goal of the automation – to save money or to advertise to your customers? Don't be swayed[4] by the 'what harm can a short message do'.

Adapting to callers

There's an argument that goes

> *'people won't learn to use the system – and why should they – the system should adapt'*

Which results in UI designers spending a great deal of time trying to make their interface adapt to whatever the caller wants (or we predict might want) to do.

Well that seemingly sensible assumption is incorrect. People are extremely good at learning and adapting – and they do it without thinking. This is good news for UI design. It means that 'all we need to do' is have a good, consistent, predictable (i.e. easy to learn) UI and callers will figure it out. Remember that real[5] callers have a significant stake in the outcome of the interaction – they want it to work.

NB: Adapting to the caller has another drawback. Callers will continuously try to predict what is coming next – by adapting, the automation is making it harder for the caller to predict. This makes it harder work for the caller.

Well I think most callers will be…

Generally you cannot categorise callers – everyone is different – it largely depends upon their 'frame of mind' and 'setting' (physical environment) – and it changes from call to call and even during the call! Don't design around a mythical demographic.

---

[3] Remember how I said people will never be 'pleased' with the interface itself?
[4] Like you'll have any say in the matter – it's a touch point and marketing gets first dibs on all those ☹
[5] As opposed to 'test callers' who have no incentive to learn/adapt (remember this fact when taking feedback)

Operator!

Always have the operator/zero function available (even if you don't voice it) – this is a consistency thing – as automation permeates daily life, people come to expect it. This doesn't mean that you have to advertise the fact (too aggressively) and it doesn't mean you've got to action the request – you can negotiate with the caller. It is important, however, to acknowledge the callers request.

Remember the goal of your organisation as a whole is to serve its customers (i.e. the callers.) There are times when only a human will do – and the rest of the time, if the automation is good enough, people will be happy to self-serve. The paradox is that the automation KPI's often favour 'containment' (keeping people away from agents) – i.e. keeping your customers at arms length?!

Hindering access to the operator doesn't stop people – it just annoys them!

Sorry I didn't get that…

Don't waste time apologising and trying to explain to the caller that there has been a problem and what you think it might be – they're not interested – you aren't sorry and you don't know what went wrong – so shut up and give them another go (rapid re-prompt.) - possibly with a targeted hint.

Apologies serve no purpose other than to waste time, confuse and irritate the caller.

…except for the odd 'sorry' – used as a 'bracing advisory'- indicating to the caller that they are probably not expecting the next prompt ("Sorry – was that book tickets?")

Remember, you can interrupt me at any time…

Be aware of the reflexive, low level almost subconscious behaviour programmed into us all. Turn taking, barging in, taking and relinquishing the floor cannot be 'instructed'.

There are some interfaces that absolutely require the caller to barge in – for example 'speech pointing' (saying "that one" during a list) or to terminate the delivery of a large amount of information ("once you've heard enough say stop") Even these types of UI don't need to specifically instruct the caller to interrupt – it's implicit.

Please visit our website at WWW

Do you really think in this day and age that a caller doesn't know about the web? What do you expect them to do – hang up and go to the web? Don't waste their time – they made the decision to use the phone – just let them! This is actually a marketing/branding issue – see above.

Practical Suggestions:

What follows are some suggested 'best practice tricks' to incorporate into your applications – nothing too prescriptive – remember every interface is different ☺:

Remember: *A good UI should go unnoticed.*

- Be predictable and consistent. Don't try to adapt to the caller – let them adapt to you - people are brilliant at learning and adapting – give them something to 'latch onto'.
- Use second person imperative ("Say your account number" rather than "Tell me your account number") Don't worry too much about being perceived as 'rude' – you are a machine – language etiquette doesn't apply so much!
- Be conservative and formal rather than colloquial – professional is the goal (nobody is annoyed by Radio4 – but many are by Radio1.)
- Keep yes/no grammars 'streamlined' – this is your general error recovery mechanism. Having 'yep' and 'nope' is going to introduce false acceptance – just when you don't want it. The only exception is 'yes please' and 'no thanks' (in English) and possibly 'sure' (in Canadian)
- Just In Time (JIT) learning. Only provide help when it is needed. Telling a caller about the 'go back' option at the start of the call wastes time and will simply be ignored/forgotten. Wait until they're having difficulty – then they will be much more receptive!
- When providing help make it targeted – don't simply dump all the help alternatives at the first sign of trouble. It's a temporal interface remember – and, by definition, the caller is already having difficulty.
- "Let the user go first" - dispense with long greetings and introductions and help and explanations – let the caller get on with accomplishing their goal – cut straight to the first question ASAP.
- Use OK and please/thanks like you would salt and pepper – to taste. They quickly become irritating.
- It is ok to 'dip' into touch tone for those tricky bits (like account number entry) then switch back into speech. This is distinct from permanently switching into touchtone due to serious background noise, line quality or 'accent'.
- Don't discount touch-tone – a speech only application is guaranteed to fail under conditions that are likely to occur! Indeed touch-tone is better than speech for:
    - Accuracy (especially in noisy environments)
    - Speed
    - Privacy

Practical Suggestions Part 2

- Always let the caller know it is an automated system
- Identify application type (and hence caller demographic)
    - B2B (business)
    - B2C (consumer)
    - B2E (enterprise)
- Success breeds success (especially on first question)

- There are 5 classes of machine spoken output as follows:
    - **Prompt** (encouraging spoke response)
    - **Feedback** (indicating what has happened)
    - **Instructions** (instructing what to say)
    - **Help** (separate coaching mode)
    - **Data** (relaying information)
- Keep instructions and prompts separate – and always play instructions before prompts – only repeat the prompt.

- Avoid tense – except for ‘was that’ or ‘did you say’
- Avoid “you could” – prefer “you can”
- Avoid “would you like” – prefer “do you want”
- Avoid ‘did you say’ – prefer ‘do you want’
- Avoid mixing nouns and verbs in list of options (use one or the other)

- Use “say” for verbatim input (say a, b or c)
- Use “state” for data input (state your PIN)
- Use landmarking (**main menu:** blah blah, **new messages:** blah blah)
- Use n-best to reject previously refused options.

- Generally, no input and no match should be handled the same – reprompt (remember that you cannot be certain why the caller didn’t speak – and even if you did hear speech, you cannot be certain if it was the callers and even then if it was directed at you – so what are you going to do differently?)

- Tones can be used to indicate
    - Prompt user to speak
    - Acknowledge response
    - Landmark
    - Feedback (rewind earcon)
    - Convey error condition

- List selection need not actually recognise (speak to select)
- Don’t be afraid of inferring (guessing) and moving on rather than trying to recover every ‘error’ (i.e. in a yes/no situation, pick the most likely and benign.
- Consider using barge in to infer cancel/go back.