# Real Users and Real Dialog Systems: The Hard Challenge for SDS

Alan W Black
Maxine Eskenazi
Language Technologies Institute
Carnegie Mellon University

# The issue in this paper

- We need Real, Live DATA
  - Assessing SDS technology requires live data
    - If you change something, the human in the loop may react differently
    - So, offline testing often will not determine whether your have made a positive change to the SDS
  - Real users react differently from paid ones
    - Paid ones accept wrong results
    - Paid ones try to game the system (speed, performance)
    - And paid ones have to be recruited … and paid!
  - Industry has real applications, but they can't share the platform or the data

# More of the issue

- Real system applications that come with a flow of real users are hard to find
- Real systems are high-maintenance
- Real systems require much attention
- Real systems imply less control over the coveted real users!

- But real systems are a treasured asset for the SDS community if they are shared!

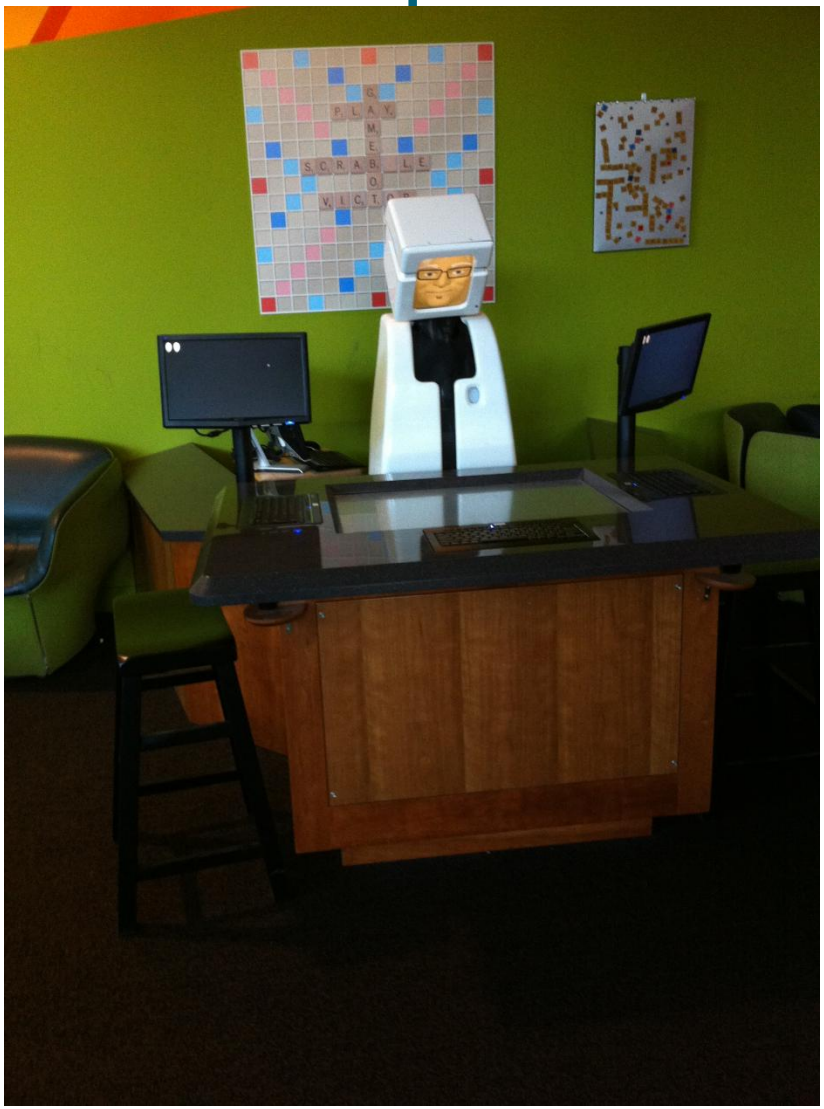
# In this talk

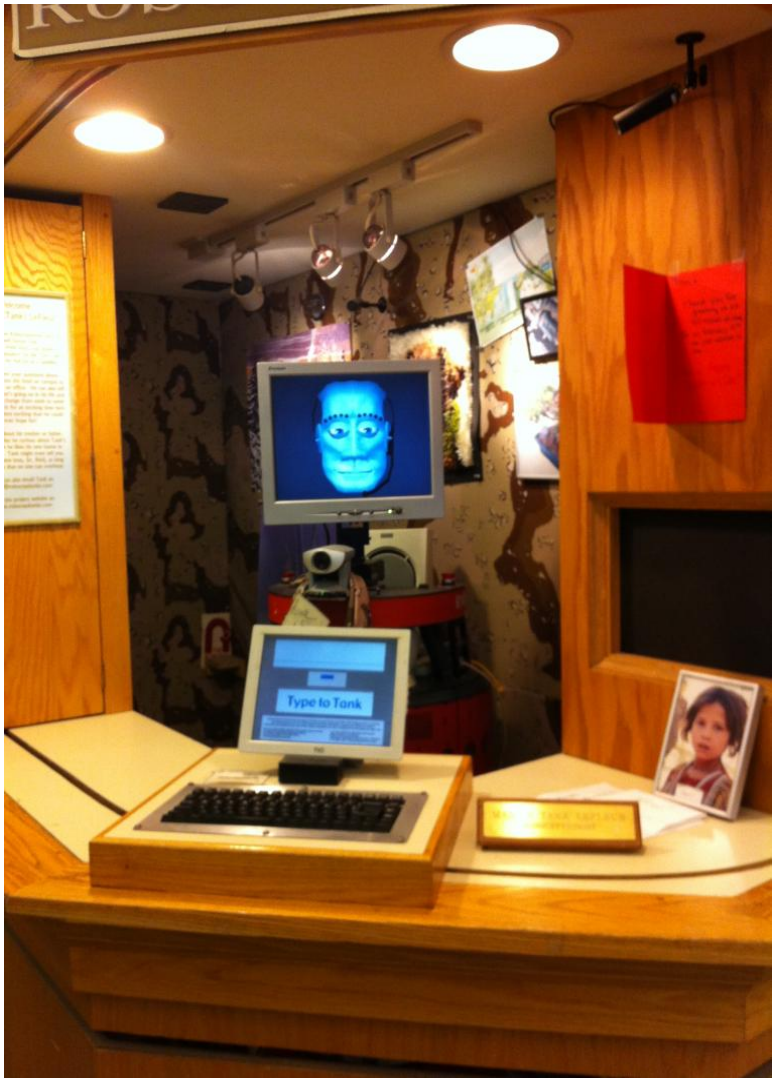
- How to find a good application
- The example of Let’s Go
- Dealing with a partner organization
- Maintaining the system

# How can you find a good application?

- To be good for research, your application should:
  - Have a ‘champion” at the partner organization you want to work with
  - Not contain personal information, if possible
  - Be something that people really need
    - And that they need to use by voice
  - Not be <u>your</u> idea of what someone may like, but rather be something people already need and use in some other way

# Looking for applications: Two examples





# Dealing with a partner organization

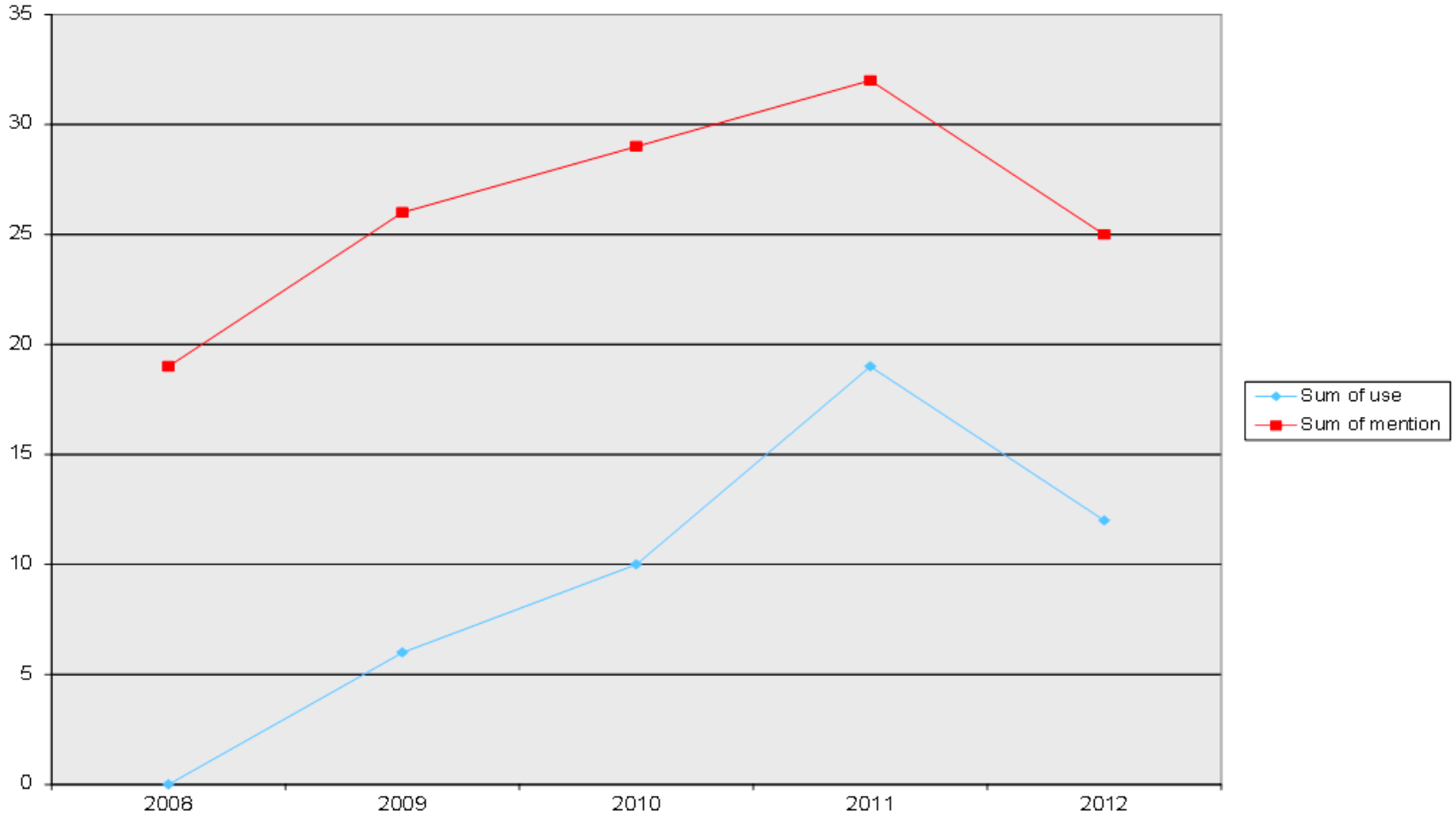
- If you find an organization to work with:
  - Show them why they need your service
  - Get a written agreement about your rights to the data
  - Determine how you will tell users that they are being recorded
  - Determine how you will maintain user privacy
  - Ask for some (even small) monetary participation in the project
  - Have a plan for what to do if your “champion” leaves the organization

# An example: The CMU Let's Go system

- Goal
  - Provide scheduling information for Pittsburgh busses
  - Nightly and on weekends
  - Gives next bus, fullness of bus, snow and other changes
- Details
  - Running daily since March 5, 2005
  - Estimated success rate is 75-80%
  - Average length of a call
    - 2007: 129 sec (~2 min)
    - 2008: 110 sec
    - 2009: 99.56 sec
- Number of dialogs so far: ~ 180,000
- About 1300 calls per month

# Your application can be simple

- Here is what Let's Go has engendered:



# Maintaining the system

- Despite appearances, the system does **not** run itself
- Start with human recordings or with WOZ
  - As soon as you have some data, retrain acoustic and language models
    - We got more from the LM retrain
    - We used Communicator data for AM
- Start with a conservative system – important - the first experience with the system MUST be successful
  - Confirm with dtmf
  - No open ended questions like “how may I help you”
  - System-directed dialog
  - Explicit confirmation
- As user confidence in the system grows, you can become less conservative

# Maintaining the system

- Ongoing maintenance:
  - Servers and other hardware
  - Automatic software updates (%&_*(@#%_)(&
  - Infrequent bug fixes
  - Constant data backup
    - And crowdsourcing pipeline to label data
  - Daily (or more often) automatic system reboot
  - Backend changes
  - Update software to keep it state of the art
  - *** test all new versions /changes thoroughly before letting them “go live”***
    - new system as good as or better than present running version

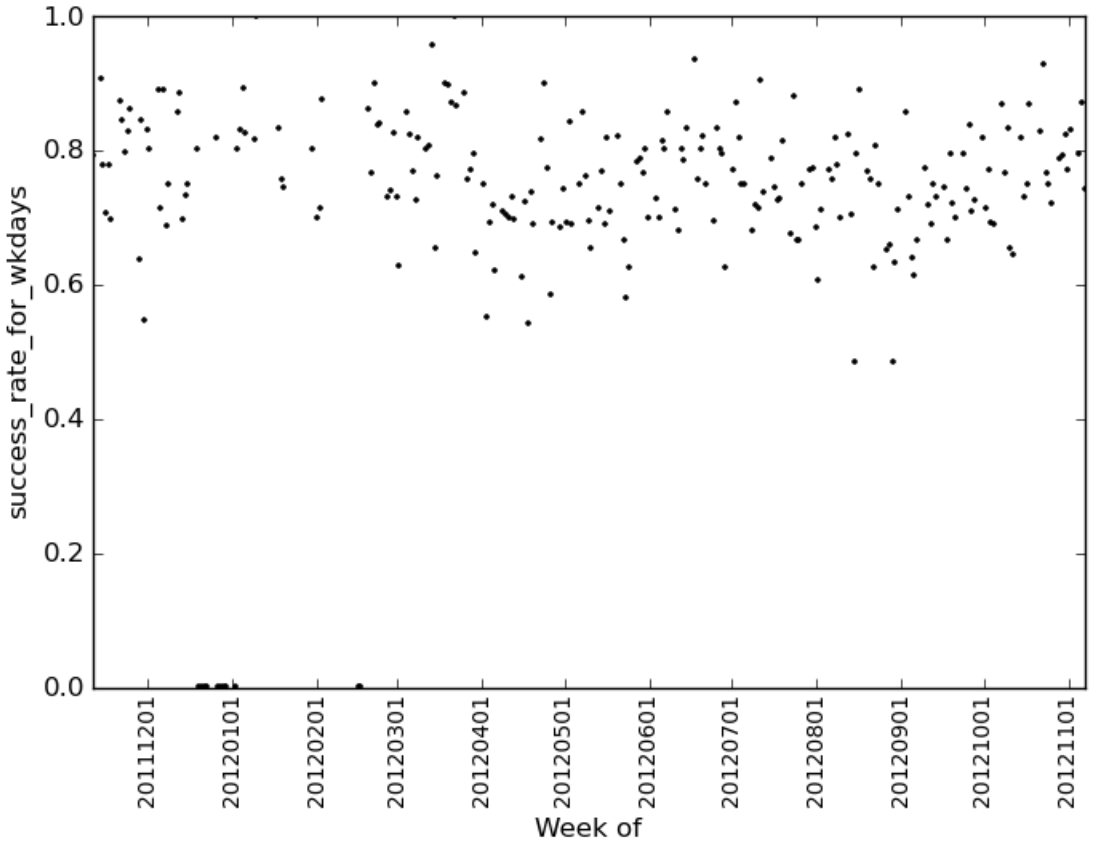
# Maintaining the system

- Detecting breakdowns
  - Automatic software updates (%*#&_*(#&%@
  - System sends email automatically upon certain failures
  - Call or use the system at regular frequent intervals
    - Someone from our group is “babysitting” the system *every* night
  - Remote restart if something is not working
  - Daily and weekly reports

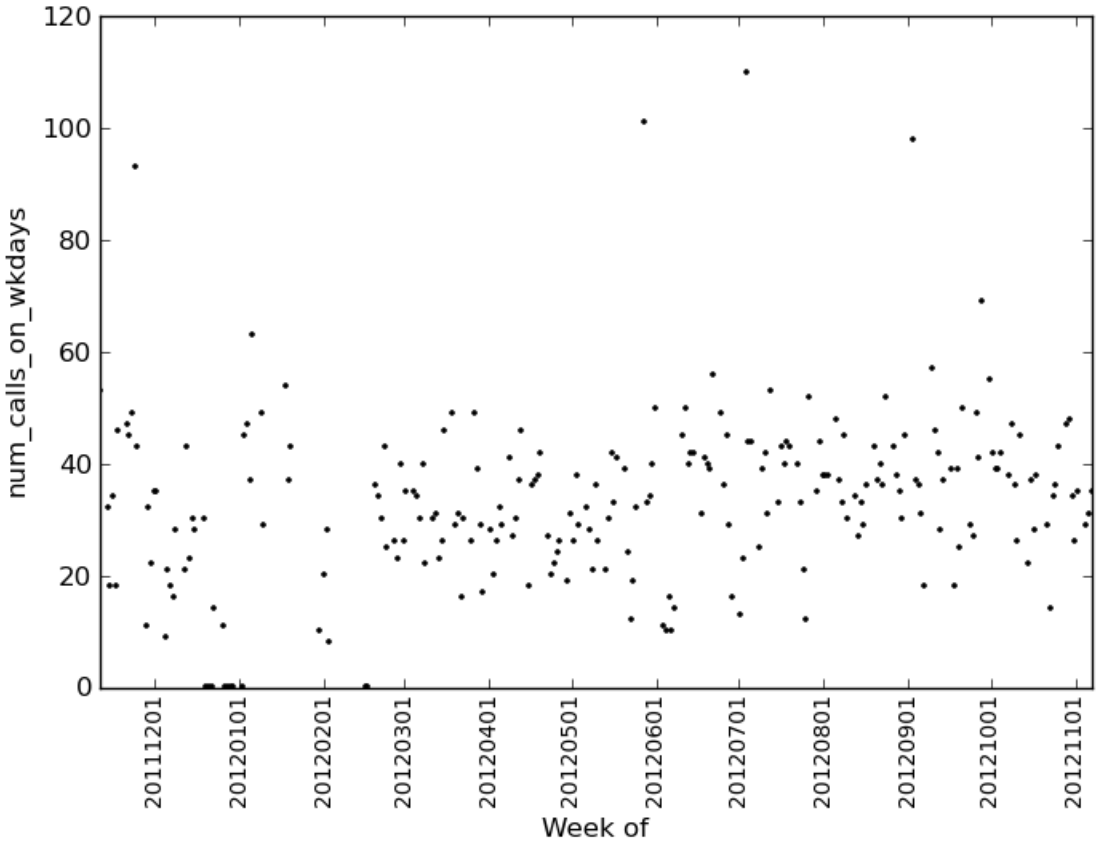
# The Let's Go daily report

- LetsGoPublic statistics for 2012-11-12
- Number of sessions: **40** [14.6-65.0] (39.8 19.37 0.01)
- Number of no-turn sessions: **6**
- Number of sessions >= 4 turns: 29 [12.2-53.2] (32.7 15.76 -0.24)
- Average number of turns per session: **10.1** [9.3-16.8] (13.1 2.89 -1.03)
- Estimated successes: 25 (**86.2 %**) [61.3-89.1] (75.2 10.68 1.03)
- View logs: http://clark.speech.cs.cmu.edu/data/LetsGoPublic2/20121112/index.html
- The numbers shown are 80% range, mean, standard deviation, and z-score, respectively, for this day of the week. Numbers are computed since April 2007, upon moving to the Olympus2 system. Potential outlier values should be noted when the z-score has a magnitude greater

# The Let's Go weekly report: success rate - weekdays



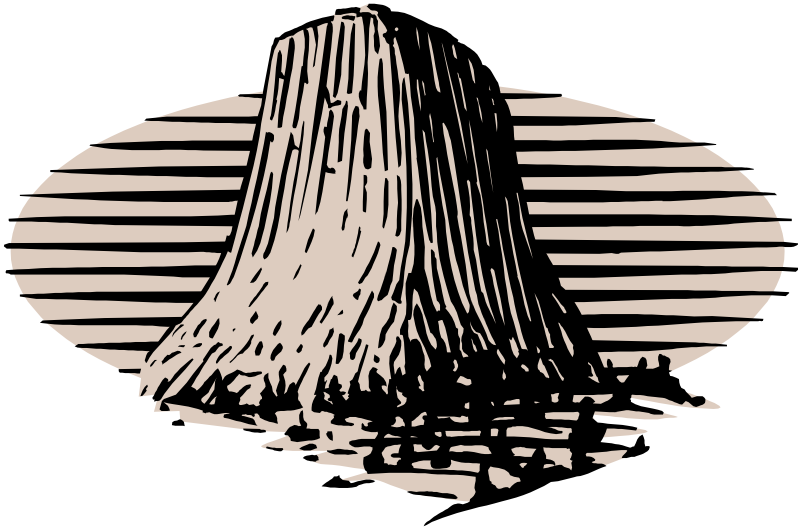
# The Let's Go weekly report: num calls - weekdays



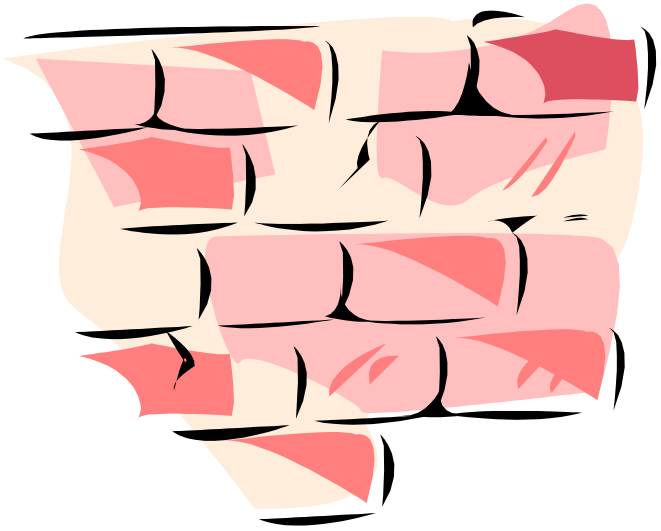
# To conclude

- Real applications with real users are very important for spoken dialog research
- The choice of the application is important
- System creation and maintenance demand much attention
- The community needs more systems that are open platforms where everyone can run studies and use the data

# Or what it's like to be between a rock and a hard place!



Need for real users and much data



Difficulty of finding real applications and users and maintaining them