RASP-IVR: A Low Cost Interactive Voice Response System

Vikram Kamath Cannanure
Carnegie Mellon University
Pittsburgh, PA
kmarkiv@cmu.edu

Timothy X Brown
Carnegie Mellon University
Kigali, Rwanda
timxb@cmu.edu

ABSTRACT

Interactive Voice Response (IVR) systems have demonstrated that they provide an opportunity for a low-literate audience to express themselves and interact with information technology systems. However, to achieve such goals in a development setting requires a low-cost system that allows participation of multiple stakeholders in the design process. In this paper, we describe a low-cost IVR hardware system called RASP-IVR that runs on a Raspberry PI and a local GSM modem. RASP-IVR is an open source appropriate system engineered to nurture community driven solutions. We have open-sourced our system and are working with a Rwandan community based organization to test our system in the field.

CCS CONCEPTS

• **Hardware** → *Sound-based input / output*; • **Applied computing** → Microcomputers;

KEYWORDS

Interactive Voice Response, Telephone Interface, Community Content

ACM Reference Format:
Vikram Kamath Cannanure and Timothy X Brown. 2018. RASP-IVR: A Low Cost Interactive Voice Response System. In *2nd African Conference for Human Computer Interaction (AfriCHI '18), December 3–7, 2018, Windhoek, Namibia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3283458.3283489

AfriCHI '18, December 3–7, 2018, Windhoek, Namibia

ACM ISBN 978-1-4503-6558-1/18/12…$15.00
https://doi.org/10.1145/3283458.3283489

1 INTRODUCTION

Interactive Voice Response (IVR) technologies[1] provide a platform for rural and low-income users in the developing world to communicate in novel ways using phone calls from their mobile phones. The voice-based interaction allows designing for users with less literacy on a ubiquitous and familiar mobile-phone interface. Prior work on IVR systems has been used for citizen journalism [26], localized education for health [31] support groups for farmers in agriculture [28], and community-based entertainment such as a music portal [23, 35], community radio [21], and viral entertainment [29]. Although these examples show that IVR systems can potentially benefit community-based organizations (CBOs) working with users in low-resource settings, a typical IVR system has high setup, call, maintenance, and staff training costs [25]. This makes it hard to invest in a system in the early stages of a project creating a hurdle to adoption. Therefore, before a full IVR investment, there is a need for a simple and low-cost system that can be used to get early feedback from the community as shown in Fig. 1.

Figure 1: RASP-IVR is a low cost ($50) IVR system consisting of a Raspberry Pi and a GSM modem.

In the past, successful IVR systems have been designed with active end-user involvement, for example, Avaaj Otalo used Wizard of Oz techniques [22], The Urban Sex Worker project used ethnography [30] and Healthline used interviews [31]. Participatory design or the concept of getting

[1] An example is the familiar airline call center: "Press one for reservations. Press two for check-in. …"

users and other stakeholders involved in the design process is a solution towards designing successful interventions in developing countries [27]. However, the earlier uses of participatory methods in IVR system design all worked at an abstract pre-implementation stage. Active user participation in working IVR prototypes can lead to more rapid and relevant solutions.

This paper describes RASP-IVR, a low-cost tool that can support early user engagement while building IVR systems. RASP-IVR is an open source appropriate system that leverages open source software and existing devices to lower deployment costs. RASP-IVR is designed for user involvement so developers, CBO staff, and other users can collaborate and learn from the local community before investing in a full solution. For small-scale deployments, RASP-IVR can also act as a low-cost long-term solution.

2 RELATED WORK

Existing IVR systems can be divided into open source, commercial enterprise, and commercial cloud solutions. Among open source systems, there are Asterisk [9] and FreeSwitch [3] which reduce the setup and airtime costs but require additional costs such as development and maintenance. Avaaj Otalo [28] was deployed in Gujarat, India to create a social network for rural farmers but needs customization and development work for other applications. Another alternative is IVR Junction developed by Vashita et al. [36]. It allows customization and has been deployed in various locations in India and Africa. It uses a local SIM card to lower calling cost but has a setup cost of around $1000 that is prohibitive at the early design stages in a low-resource setting. Commercial systems like Awaaz.De [2], Gram Vaani [5] and Voto [4] among others provide robust and customizable enterprise solutions which also require significant early investment. Cloud-based services like Twilio [14], Torpo [13], Sinch [11], and Pilvo [12] require no upfront hardware or software but have service fees. They offer APIs to build applications using various programming languages using skilled developers. In many cases, they require international calls to be paid for by the community users or the service providers. The systems often have text-to-speech and other features designed for English and other common languages that do not cater to less-common local languages. So, although existing systems provide significant benefit, they involve upfront costs, service fees, or insufficient localization to support early user involvement in low-resource settings.

We build on prior work to support active community-driven solutions in IVR. From a technical perspective, our solution is a cost-effective and simpler version of the traditional IVR system using Asterisk [9] and a router. We improve on these solutions through our low-cost hardware based on a Raspberry PI and a GSM modem to build the IVR systems.

3 RASP-IVR DESIGN

Our motivation for RASP-IVR was from our preliminary interviews with CBOs across the world interested in IVR technology. Firstly, our partners were concerned with the setup cost and were interested in a prototype before an investment in IVRs. Secondly, existing cloud systems allowed researchers to prototype applications but the power dynamic [27] was skewed towards researchers and away from participants. We reflected on these tensions [19] and saw an opportunity to build low-cost IVR systems. Lastly, following the vision of Brewer et al. [17] to promote open source in the development domain and from prior successes (such as Open Data Kit [18] and Commcare [33]) we have open-sourced our application.

The "RASP-IVR" name was chosen because it was built on the Raspberry Pi, a low-cost, Linux-based, single board computer. The OS is based on RASPBX [1] with a few customizations and libraries to better support IVR interactions. The system consists of a Raspberry Pi model 3 and a GSM mobile telephone modem. The GSM modem interfaces with an Asterisk server. The Asterisk server can be managed using FreePBX web interface to create and customize the IVR tree or voice over IP (VoIP) parameters. We have added scripts using open source libraries to extend RASPBX to support Voice messages, SMS, and local SIP transfers.

Figure 2 shows the working of RASP-IVR. The GSM modem interfaces with voice calls and text messages to connect it to an Asterisk server. The Asterisk server can connect the GSM call to the local or global VoIP network based on the requirements. The VoIP calls can be routed to various extension based on the clients touch tone input or answered through SIP clients like Yate[16], Zoiper among others using a desktop or a mobile device through a local SIP account. Additionally, the VoIP call can be transferred to external services like Twilio to use their cloud-based services e.g. Speech to text engine or cloud storage if needed. However, RASP-IVR is self-sufficient for participatory design or for small-scale deployments.

RASP-IVR is a very low-cost system and it consumes very little power. Our system can be set up for $50, i.e. $35 for a raspberry PI and $15 for the GSM Modem. The system's capability can be enhanced by connecting two or more modems allowing for more concurrent calls and call routing. Adding multiple modems may cause the modems to restart during high power consumption actions like calls. This limitation can be overcome using a low-cost USB hub which costs $10. The basic system costs $50 but the system can be augmented with low-cost tools to boost its capability based on context. For example, a 5V DC uninterruptible power supply costing $20 can be added to handle power outages.

Developing countries are not well supported by cloud-based services and even when services are present, they fall

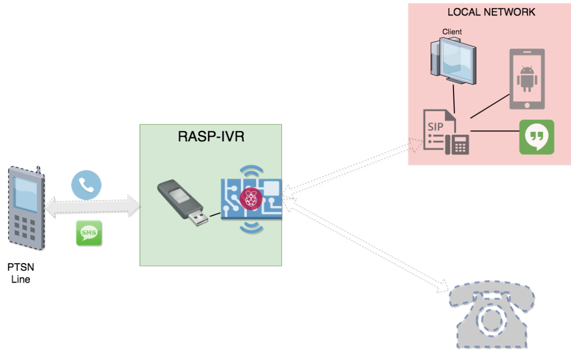

Figure 2: The RASP-IVR system can interact with GSM lines via voice calls or text messages. These interactions can be transferred to the RaspberryPI server and can be transferred to VoIP or external lines based on the application design.

beyond the budget of local communities. RASP-IVR circumvents this issue by using local SIMs which allows the use of airtime bundles and existing SIM cards familiar to users. Table 1 shows the comparison of 1000 minutes of Twilio's cloud-based usage fees versus 1000 minutes of the default local calling costs to the RASP-IVR for a few countries in East Africa and we show the percentage savings. Using a plan or airtime bundle considerably lowers the calling costs further (e.g. see Tanzania), this can be capitalized to further increase savings.

We profiled a RASP-IVR instance using automated calls using sipP. sipP [6] is a command line tool to simulate VoIP calls on so-called SIP servers. We set up a SIP channel on a RASP-IVR instance (on an MTN E153 modem and Raspberry PI 3) and tested the system using a sipP script. We ran a simulation for 10 calls per second, 20 concurrent calls, and 100 total calls and measured the response rates. A test call dialed into a SIP channel and listened to a recorded message, We found that RASP-IVR successfully handled all 100 of the calls with no failure in 2 mins. The system handled 0.828 calls/second. As the experiment progressed, the response times of calls converged towards 0.015 seconds. Figure 3 shows the plot of response times over the duration of the experiment. This shows the system VoIP portion of the system can handle simultaneous calls with little delay in call setup. Note that unlike the VoIP interface, the GSM interface dongle can handle only one call at a time. Additional dongles can be added to build capacity, although each dongle would have a different phone number.

Table 1: Comparison of airtime cost for RASP-IVR compared to Twilioś (Cloud) airtime charges for 1000 mins for countries in East Africa

Country	Twilio	RaspIVR(Local)	Saving%
Kenya	$106	$70[10]	34%
Uganda	$430	$64[8]	85%
Rwanda	$405	$52[7]	87%
Tanzania	$500	$22(plan)[15]	96%

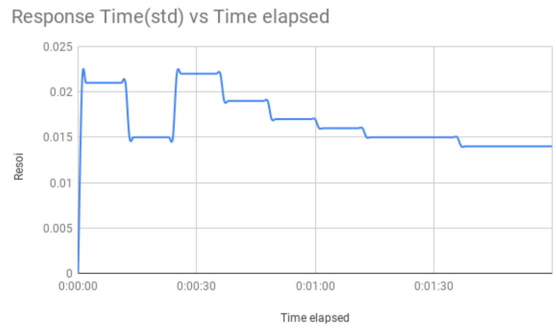

Figure 3: Shows response times for RASP-IVR as time progresses for 10 calls per second, 20 concurrent calls and 100 total calls. This tells us the responsiveness of the RASP-IVR for handling concurrent calls. Although the number of calls is limited to the number of modems, the system can route the call internally and handle complex scenarios such as conference calls.

4 DISCUSSION

RASP-IVR opens the possibility of creating custom applications that can support contextually relevant challenges using its voice and text messaging platforms. The RASP-IVR enables features such as automatic redirecting of calls, user selected content, recording voice content from callers, text message-based interaction, data collection, and personalized content based on caller ID. However, we are reflective that IVR based systems like all technology needs to be designed to magnify existing human development efforts [34].

While there are caveats discussed further below, advantages of IVR-based systems include the following. A purely voice-based system appeals to low-literate users as compared to text-message or web-based applications. IVR systems are accessed through the already familiar process of calling on widely available mobile phones. The mobile phone access can be more private and less stigmatizing than physical visits to sensitive services such as mental health counseling or family planning. Caller ID allows the IVR system to call back dropped calls and to connect activity from the same phone across different calls. Lastly, the voice signal itself contains useful information about the speaker including estimates of age, weight, gender, stress, and other health factors [32]].

We have been working with a few partners in the health sector in Rwanda. Our partners have existing face-to-face operations and we hope to build systems to extend the coverage and the reach of their services [34]. IVR can reach clients who cannot travel to existing centers providing extensions, albeit limited, of their existing services. Interestingly, one NGO partner periodically travels to outlying villages and views the IVR as a tool for driving visitors to these outreach activities. The IVR can reduce the workload of staff through automated services that cover outside of normal hours or collect and analyze user data. One partner will use this feature to automate reporting to donor sponsors. Another partner views the IVR as part of a larger transition from paper to electronic health records (EHR). The IVR could be integrated with the EHR system [24] and a web application where the IVR collects caller data which prompts the web application to pull up the client EHR before speaking with a counselor. The call is followed up with a text message to the client and another message to the field office closest to the client.

Against these benefits, there are, however, some caveats. The IVR system is language specific and prompts need to be recorded in the different languages and dialects of the user population. While the phone interface is familiar, IVR systems are inhibited by the social and cultural norms of developing communities. Users more familiar with face-to-face interaction may not be receptive to automated interactions and data queries. Indeed, one partner anticipates that they will get the best response after their clients are shown how to use the IVR system during face-to-face meetings. We also found that access to phones themselves are a limitation for low-resource users. Phones are not always charged due to cost and availability of electricity. Many times phones are shared so that the same phone might be used by different users and a given user might use different phones. Follow up messages and calls may not reach the intended person. Further, the use of a shared phone can prompt interest and suspicion about who is being called. Finally, the IVR systems raise a number of privacy concerns from how collected data is stored, transmitted, and reported to how one user is prevented from accessing another's data.

From the developer side, the current system requires programming expertise which may not be readily available for a CBO. We found that not all modem dongles worked with the system and it was trial and error to find a model that worked correctly. In some cases, it was a hardware issue where the dongle would not communicate with the Raspberry Pi. In others, it was a network issue: One service provider did not transfer the touch tone signals in mobile-to-mobile calls (including calls to the GSM dongle). The system, at present, can handle one call at a time and is not appropriate for high-calling volume deployments. For our partners who anticipate modest volumes the RASP-IVR is appropriate. If volume were to grow, they could justify a transfer to one of the more scalable solutions in Section 2.

Future Work

We have presented RASP-IVR a low-cost open-source system to design IVR systems with the local communities in the developing world. We expect to extend our current design in a field evaluation. We are working with Rwandan community based organizations to test our system in the field. We expect this to open new directions for RASP-IVR and also help correct existing assumptions we have in our design.

We expect RASP-IVR to act as an IVR plugin to Open Data Kit [18] and serve as an alternative to the deprecated ODK Voice [24]. For data management we see opportunities to integrate tools like ODK Aggregate [18] or Open MRS [20].

We will be releasing RASP-IVR with sample code and detailed tutorials for people to set up IVR systems. We hope to create a web platform for community-driven RASP-IVR solutions. Our work in progress portal can be found at

http://bit.ly/rasp-ivr.

5 CONCLUSION

Interactive Voice Response (IVR) systems have demonstrated that they provide an opportunity for a low-literate audience to express themselves and interact with information technology systems. However, IVR systems have operational cost and setup overhead creating a barrier to entry. In the past, successful IVR systems have been designed with active end-user involvement. However, to achieve such goals in a development setting requires a low-cost system that allows participation of multiple stakeholders in the design process. In this paper, we have described RASP-IVR a low-cost and low power IVR system to help build community-driven solutions for the developing world. RASP-IVR is an open source appropriate system designed for early participatory development before transition to robust large-scale deployments. The cost effective nature of RASP-IVR allows room for closer and deeper stakeholder engagement to pilot early-stage systems before investing in a full-fledged IVR solution. We have open sourced our system and continue to develop our tool through a field evaluation with a Rwandan community based organization. While focused on CBOs in this paper, RASP-IVR is useful for local entrepreneurs and outside organizations (such as NGOs) to develop and test IVR solutions as well.

ACKNOWLEDGMENTS

We would like to thank Placide Hakuzweyezu and Saphani Bazimya from the Fablab Rwanda for sharing their modem and brainstorming ideas during development. We also thank our research assistants Justin Bigwi and Nebiyou Yismaw who helped verify the airtime numbers. We are very grateful to the Human Computer Interaction Institute at Carnegie Mellon and the CMU Africa campus for supporting this work.

REFERENCES

[1] [n. d.]. Asterisk for Raspberry Pi. http://www.raspberry-asterisk.org/. (Accessed on 03/26/2018).

[2] [n. d.]. Awaaz.De | Mobile Solutions for Social Impact. https://awaaz.de/. (Accessed on 03/02/2018).

[3] [n. d.]. FreeSWITCH. https://freeswitch.org/. (Accessed on 03/02/2018).

[4] [n. d.]. A global social enterprise improving lives via mobile - Viamo. https://viamo.io/. (Accessed on 03/02/2018).

[5] [n. d.]. gramvaani | community-powered-technology. http://www.gramvaani.org/. (Accessed on 03/02/2018).

[6] [n. d.]. mojolingo/sippy_cup: Create SIP load test scenarios the easy way. https://github.com/mojolingo/sippy_cup. (Accessed on 07/13/2018).

[7] [n. d.]. MTN Rwanda - Welcome. http://www.mtn.co.rw/Content/Pages/40/MTN_Per_Second. (Accessed on 07/13/2018).

[8] [n. d.]. MTN Uganda | Products | MTN Call Per Second. https://www.mtn.co.ug/en/products/tariff-plans/Pages/mtn-call-per-second.aspx. (Accessed on 07/13/2018).

[9] [n. d.]. Open Source Communications Software | Asterisk Official Site. https://www.asterisk.org/. (Accessed on 03/02/2018).

[10] [n. d.]. Safaricom Rates | ILRI/ICRAF ICT Portal. http://ict.worldagroforestry-ilri.org/safcom. (Accessed on 07/13/2018).

[11] [n. d.]. Sinch | Voice API, Verification & Cloud VoIP made Simple. https://www.sinch.com/. (Accessed on 03/26/2018).

[12] [n. d.]. SMS API, Voice API, & Global Carrier Provider | Plivo. https://www.plivo.com/. (Accessed on 03/26/2018).

[13] [n. d.]. Tropo: Home. https://www.tropo.com/. (Accessed on 03/26/2018).

[14] [n. d.]. Twilio - Communication APIs for SMS, Voice, Video and Authentication. https://www.twilio.com/. (Accessed on 03/07/2018).

[15] [n. d.]. Voice & SMS Overview. https://vodacom.co.tz/en/voice-and-sms-overview/. (Accessed on 07/13/2018).

[16] [n. d.]. Yate. http://yate.ro/. (Accessed on 07/13/2018).

[17] Eric Brewer, Michael Demmer, Bowei Du, Melissa Ho, Matthew Kam, Sergiu Nedevschi, Joyojeet Pal, Rabin Patra, Sonesh Surana, and Kevin Fall. 2005. The Case for Technology in Developing Regions. *Computer* 38, 6 (June 2005), 25–38. https://doi.org/10.1109/MC.2005.204

[18] Waylon Brunette, Mitchell Sundt, Nicola Dell, Rohit Chaudhri, Nathan Breit, and Gaetano Borriello. 2013. Open Data Kit 2.0: Expanding and Refining Information Services for Developing Regions. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (HotMobile '13)*. ACM, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/2444776.2444790

[19] Kim VL England. 1994. Getting personal: Reflexivity, positionality, and feminist research. *The Professional Geographer* 46, 1 (1994), 80–89.

[20] Suranga Nath Kasthurirathne. 2012. Designing and Developing an Open Source Medical Informatics Module. *XRDS* 19, 2 (Dec. 2012), 20–23. https://doi.org/10.1145/2382856.2382866

[21] Konstantinos Kazakos, Siddhartha Asthana, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, Amarendar Reddy Manindla, Subhashini Arcot Manikam, GVS Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, and Patrick Olivier. 2016. A Real-Time IVR Platform for Community Radio. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 343–354. https://doi.org/10.1145/2858036.2858585

[22] J. F. Kelley. 1984. An Iterative Design Methodology for User-friendly Natural Language Office Information Applications. *ACM Trans. Inf. Syst.* 2, 1 (Jan. 1984), 26–41. https://doi.org/10.1145/357417.357420

[23] Zahir Koradia, Piyush Aggarwal, Aaditeshwar Seth, and Gaurav Luthra. 2013. Gurgaon Idol: A Singing Competition over Community Radio and IVRS. In *Proceedings of the 3rd ACM Symposium on Computing for Development (ACM DEV '13)*. ACM, New York, NY, USA, Article 6, 10 pages. https://doi.org/10.1145/2442882.2442890

[24] Adam Lerer, Molly Ward, and Saman Amarasinghe. 2010. Evaluation of IVR Data Collection UIs for Untrained Rural Users. In *Proceedings of the First ACM Symposium on Computing for Development (ACM DEV '10)*. ACM, New York, NY, USA, Article 2, 8 pages. https://doi.org/10.1145/1926180.1926183

[25] Eleanor R. Marchant. 2016. *Interactive voice response and radio for peacebuilding: A macro view of the literature and experience from the field*. Technical Report. Annenberg School of Communications University of Pennsylvania. http://www.global.asc.upenn.edu/app/uploads/2016/02/IVR-Lit-Review_Final.pdf.

[26] Preeti Mudliar, Jonathan Donner, and William Thies. 2012. Emergent Practices Around CGNet Swara, Voice Forum for Citizen Journalism in Rural India. In *Proc. of the Fifth International Conference on Information and Communication Technologies and Development (ICTD '12)*. ACM, New York, NY, USA, 159–168. https://doi.org/10.1145/2160673.2160695

[27] Peter Oakley. 1991. *Projects with people: The practice of participation in rural development*. International Labour Organization. 284 pages.

[28] Neil Patel, Deepti Chittamuru, Anupam Jain, Paresh Dave, and Tapan S. Parikh. 2010. Avaaj Otalo: A Field Study of an Interactive Voice Forum for Small Farmers in Rural India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 733–742. https://doi.org/10.1145/1753326.1753434

[29] Agha Ali Raza, Mansoor Pervaiz, Christina Milo, Samia Razaq, Guy Alster, Jahanzeb Sherwani, Umar Saif, and Roni Rosenfeld. 2012. Viral Entertainment As a Vehicle for Disseminating Speech-based Services to Low-literate Users. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development (ICTD '12)*. ACM, New York, NY, USA, 350–359. https://doi.org/10.1145/2160673.2160715

[30] Nithya Sambasivan, Julie Weber, and Edward Cutrell. 2011. Designing a Phone Broadcasting System for Urban Sex Workers in India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 267–276. https://doi.org/10.1145/1978942.1978980

[31] J. Sherwani, N. Ali, S. Mirza, A. Fatma, Y. Memon, M. Karim, R. Tongia, and R. Rosenfeld. 2007. HealthLine: Speech-based access to health information by low-literate users. In *2007 International Conference on Information and Communication Technologies and Development*. 1–9. https://doi.org/10.1109/ICTD.2007.4937399

[32] Rita Singh, Bhiksha Raj, and Deniz Gencaga. 2016. Forensic anthropometry from voice: an articulatory-phonetic approach. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2016 39th International Convention on*. IEEE, 1375–1380.

[33] T Svoronos, P Mjungu, R Dhadialla, R Luk, C Zue, J Jackson, and N Lesh. 2010. CommCare: Automated quality improvement to strengthen community-based health. *Weston: D-Tree International* (2010).

[34] Kentaro Toyama. 2011. Technology as amplifier in international development. In *Proceedings of the 2011 iConference*. ACM, 75–82.

[35] Aditya Vashistha, Edward Cutrell, Gaetano Borriello, and William Thies. 2015. Sangeet Swara: A Community-Moderated Voice Forum in Rural India. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 417–426. https://doi.org/10.1145/2702123.2702191

[36] Aditya Vashistha and William Thies. 2012. IVR Junction: Building Scalable and Distributed Voice Forums in the Developing World. In *Presented as part of the 6th USENIX/ACM Workshop on Networked Systems for Developing Regions*. USENIX, Boston, MA.