

Experiences of Deploying and Commercializing a Community Radio Automation System in India

Zahir Koradia[§]
Dept. of CSE, IIT Bombay
Mumbai, India

Balachandran C
Gram Vaani[§]
New Delhi, India

Kapil Dadheech
Gram Vaani
New Delhi, India

Mayank Shivam
Gram Vaani
New Delhi, India

Aaditeshwar Seth[§]
Dept. of CSE, IIT Delhi
New Delhi, India

ABSTRACT

Community Radio (CR) stations are short range FM stations that seek to meet the needs of their listeners by involving them in content creation, feedback, and station management. In India, the community radio movement took off in 2008 when the first set of CR stations became operational. During that period we began design and development of the Gramin Radio Inter-Networking System (GRINS), an automation system designed to meet the ICT needs of CR stations. Over the last 2.5 years we have commercialized GRINS and deployed it at 19 CR stations in India and South Africa. In this paper, we present our experiences of software feature design, installation, training, technical support, and commercialization of GRINS in India.

We demonstrate a number of aspects: Building a small feature set at first, and then adding features on-demand ensures that utility of the software is maintained and imposition of developers' assumptions is avoided. The common notion of keeping costs low for developing regions should not come at the sacrifice of stability of the solution. Standardization of deployments, even at the loss of flexibility for the customers, is important to keep the costs of technical support low. Through technical support data collected over one year, we show that physical moving parts are easy targets to introduce faults. Although identification of most faults can be done over the phone, remote Internet access is needed to resolve many faults. Finally, we highlight that selling to NGOs is hard; the best way is to club the sale as a part of a larger project for which the NGO is being funded. We believe these insights will be useful to guide others who want to commercially provide ICT solutions in developing regions.

1. INTRODUCTION

Community Radio (CR) stations are short range FM radio stations that cater to the information needs of communities

living in the surrounding areas. With programs centered around discussions on local civic amenities, health and hygiene, advice on common economic activities such as agriculture, local folk songs, and cultural events, CR stations focus on the social development of the community they serve. In India, the government permitted non-profit organizations and education institutions to apply for CR licenses in 2006, after which a significant increase in the number of CR stations was expected.

In early 2009, we surveyed ten CR stations in India to explore the potential of using information and communication technologies to make the CR stations more effective. Our survey revealed that CR stations in India were using an ad-hoc mix of technologies leading to in-efficient use of staff time. They used a basic media player for broadcast making it hard for them to automatically schedule playlists or plan for the radio jockey to speak on air. They also did not have a solution for management of content created by the staff. Our survey also showed that in spite of high penetration of mobile phones in India, very few stations had utilized telephony to encourage community participation, mainly due to high costs of analog taps typically used in radio stations.

To address these challenges, we designed and developed the Gramin Radio Inter-Networking System (GRINS), an automation system that allows stations to schedule broadcasts, speak live on air, manage content, make and receive phone calls, and diagnose software and basic cable faults [6]. Our long-term aim was to build GRINS as a product that could be sold to CR stations, allowing all GRINS related activities to be self sustaining or even profitable. To this end, we piloted GRINS at three CR stations between June 2009 and March 2010 to design and revise several initial features. Between April 2010 and October 2011, we installed GRINS at 13 additional CR stations in India and 3 in South Africa, most of which were paid installations. During these pilots and installations we gained several insights regarding feature design, installation utilities and hardware, training modules, technical support tools, and sales strategies. In this paper, we share these insights, which we hope will be useful for others attempting to design scalable ICT solutions for developing regions.

Over the past 2.5 years of GRINS design, deployment, and commercialization we have learned several things:

1. When designing for large scale adoption, it is advis-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM DEV'12, March 11-12, 2012, Atlanta, GA

Copyright 2012 ACM 978-1-4503-1262-2/12/03 ...\$10.00.

able to start with a few essential features identified through surveys with a variety of potential customers, and build them well to gain confidence of the customers. It is important not to impose our understanding of feature needs on the customers.

2. While cost is a concern in developing regions, stability of hardware/software is even more critical to ensure quality and reliability.
3. Standardization of hardware and software configurations, even at the loss of flexibility, makes it easier to provide technical support, particularly when there is no Internet connectivity.
4. When selling solutions to non-profit organizations, it is best to make a sales pitch when the organization is initiating a project. It becomes hard for NGOs to justify expenses outside their approved scope of budget once the project is already underway.

We discuss more details of these aspects in the rest of the paper. In the next section, we briefly introduce the design of GRINS, followed by a discussion about the evolution of GRINS features in Section 3. We then present our experiences related to GRINS installation and training in Sections 4 and 5 respectively. Section 6 presents our learning in providing technical support to CR stations and Section 7 describes our experience in selling GRINS to NGOs. Finally, we conclude the paper in Section 8.

2. GRINS: AN AUTOMATION SYSTEM FOR COMMUNITY RADIO

The basic tasks carried out at a typical community radio station are to create content, schedule it for broadcast, manage it to enable efficient search and reporting, and interact with the community to gather feedback and suggestions. Below we describe each of these activities in brief to set the context for explaining GRINS features:

Content creation: Station staff go out in the field to record interviews and stories from the community. Folk songs and radio dramas are also recorded in the studio. The field and studio recordings are then edited to prepare it for broadcast.

Broadcast: The station staff create playlists of audio prepared during the content creation process, and schedule them for broadcast. The playlists may also include slots to speak live on air or take phone calls.

Content management: The staff have to manage content created at the station or recorded over phone calls, to be able to retrieve it later. They also want to track broadcast statistics to be report their activities to donor agencies and higher management.

Community interaction: Community interaction at CR stations is always an on-going effort to involve the community in various activities of the CR station, such as to solicit feedback about programs, entertain requests for repeat telecasts, hold live phone conferences and discussions, etc.

GRINS is an automation system for community radio stations that makes many of these tasks easier. It provides a single console to perform a large variety of operations, including:

- play out programs on air,
- preview programs while another program is playing,
- speak live on air and record the live speech,
- receive and make phone calls, record conversations (online telephony),
- record voice messages from callers when telephony is not in active use (offline telephony),
- add/edit metadata for programs,
- search for programs using faceted metadata search [5],
- automatically diagnose cable and software faults, and
- stream broadcast content over the Internet

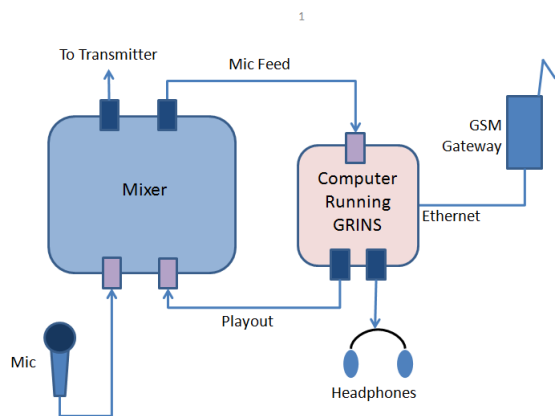


Figure 1: A radio station setup using GRINS: The computer running GRINS is connected to the mixer, which is in turn connected to the FM transmitter. The GSM gateway allows GRINS to make and receive calls.

Figure 1 shows the schematic of how GRINS typically plugs into a radio station. The GRINS playout soundcard and an external microphone connect to the inputs of an audio mixer for broadcast, and the combined mixer output goes to the FM transmitter. A duplicate of the mic feed is also brought into GRINS for archival. The headphones are used to preview audio when something else is being broadcast.

A GSM gateway is used to convert GSM calls to SIP calls and vice-versa, thus allowing GRINS to make and receive calls. The operator can pick up the call through the GRINS console itself, and talk to the caller using the same set of headphones as those used for preview. The call is automatically archived, and can also be put live on air by redirecting both the incoming and outgoing call legs to the playout soundcard in GRINS. A screenshot of GRINS user interface is shown in Figure 2.

GRINS was first deployed in July 2009 at Radio Bundelkhand, a CR station located in central India. It was revised several times based on feedback from the station staff. We then piloted GRINS at two more stations in February and March 2010. Our first paid installation took place in June 2010 at Kumaon Vaani, a station located in the Himalayas.

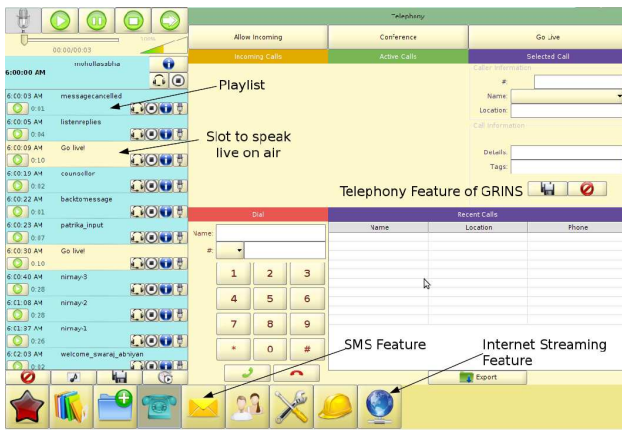


Figure 2: A screenshot of GRINS GUI.

Since, then we have deployed GRINS at 12 other CR stations in India, all of which were paid for by the CR stations. In September 2011, we piloted GRINS at 3 CR stations in South Africa. Figure 3 shows a time-line of GRINS deployments in the two countries.

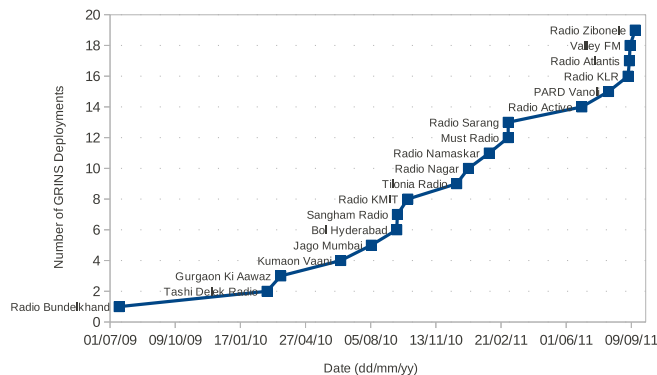


Figure 3: A time-line of GRINS deployments.

Based on recently collected statistics from CR stations, a total of more than 100 hours of audio is broadcast daily using GRINS, and about 4 hours of live speech is archived by GRINS every day. In addition, more than 1200 phone calls are made/received every month via GRINS, and 4 of the 19 CR stations use GRINS to stream their broadcast content over the Internet. We next present our experiences in designing GRINS features.

3. FEATURE DESIGN

Like any other software development project, GRINS evolved in stages with the addition of new features demanded by the stations, redesigning certain old features, and creating some breakthrough features that had not been anticipated in advance by the stations. The unique aspect in this iterative development though was the feature-solicitation policy we followed at different stages, albeit somewhat unconsciously.

We next describe the different iterations of GRINS to bring out this aspect.

Start with doing important things well

In the first iteration of GRINS, which was only deployed at Radio Bundelkhand, we started with just a few features revealed in our survey as being essential to CR stations. This included the archival of live speech, playlist scheduling, a content management library, and software diagnostics. We did this to make sure that we built something that would surely be useful from the onset. Among the features we built, playlist scheduling was the most used and eventually became one of the most attractive features of GRINS. Being an important feature of immediate utility, the playlist scheduler also quickly drew attention during any demonstrations of GRINS and helped differentiate it from generic media players as they do not have a scheduling feature by default.

However, we were not entirely accurate in interpreting the survey results, which was a result of our own biases. During our visits to the deployments, we realized that the library was not used because we had included pre-defined categories while stations wanted to create their own categories. We explain later that in spite of redesigning the library feature, it is not extensively used across all our installations. Diagnostics to test for disk space and network connectivity were also not used because, as described later in Section 6, most problems were related to cable mis-configurations and software bugs.

Choose good pilot locations and gain their confidence

In the second iteration, we fixed several bugs discovered in GRINS and reworked the features based on feedback from Radio Bundelkhand. We completely redesigned the library component: now stations could create their own categories/tags, make hierarchies of tags, and associate content with entries for staff and community members in a contacts database. We also did small tweaks to make the user interface more intuitive, such as indicate the status of broadcast being on air or the archiving of live speech in progress. This of course improved GRINS, but a major reason for putting significant time and effort was to gain the confidence of our initial set of users. This held us in good stead in the long run in terms of referrals for new installations, and an indication of commitment from our end. Upon seeing our commitment, the staff also patiently coped with the inefficiencies and bugginess in the initial versions.

In that sense, the choice of our pilot location at Radio Bundelkhand turned out to be very appropriate; the staff were immensely cooperative and helpful. We were also able to collect quality feedback due to keen observations made by one technical staff member who had prior experience of working with the national radio. Support from partner organizations for pilots is certainly crucial for the pilot's success.

Go with the demand

Once we had a few dedicated users and our deployments started expanding, we got more data points about which features would be useful for a large set of customers. Figure 4 shows GRINS usage derived from usage statistics collected by instrumenting GRINS at 10 CR stations.¹ While

¹For each feature, usage statistic is compared with our expectation of its usage to arrive at the label of 0%, 50%, or 100% usage, which is then averaged across all stations. For

live speech archiving, playlist scheduling, and telephony are frequently used, we see very little use of the library feature. Below we detail how going with the demand as against building features based on our assumptions was the right way of building features in GRINS.

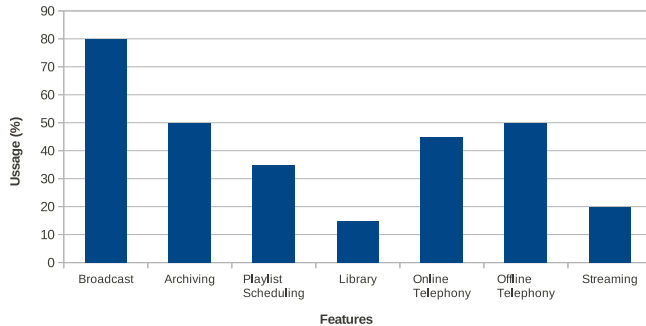


Figure 4: Feature-wise distribution of GRINS usage at 10 CR stations.

Library: GRINS usage studies at our pilot deployments and later across other deployments showed that we had misread the need for a good library interface. We were driven by the assumption that stations would want to organize their content so that they could retrieve it easily and generate reports if they want statistics (for example) on how many women related programs did they broadcast in the last month, etc. However, the library feature is actively used in only 3 of our deployments.

Even though this requirement for content management came up during our initial survey, in hindsight we feel that the error happened because we were used to meticulous file management in our personal lives and saw the station’s practices as very messy. In actuality, stations do not need that level of content management because they rarely refer back to old content for retrieval. Most stations instead choose to stick to the familiar Windows explorer interface and maintain their content on an external hard-drive; they add the content to GRINS to make daily playlists and do not refer to it again from within GRINS. The requirement for daily or monthly reports also seems to depend on how much control the parent organization exercises in demanding reports. In essence, we had imposed the library feature on the station, which eventually did not get used.

Online Telephony: While we were building GRINS, the community radio movement in India was also picking up and stations were experimenting with new techniques of community participation, including mobile phones. An inspiring story was how a station reporter attending a Panchayat (local governing body in Indian villages) meeting relayed the meeting discussion over his cellphone to the station, and the station placed their speakerphone adjacent to the microphone to broadcast the meeting live on air. This could revolutionize transparency in local village governance! Based on de-

example, 8 stations always use GRINS for broadcasting, and 2 stations use CD players for the same. Labelling 100% for 8 stations and 0% usage for the other two we arrive at 80% usage of the broadcast feature.

mand from the stations and seeing its impact potential, we built a telephony component in GRINS through which station operators could pick up calls and put them live on air. Soon after, telephony began to be used in interesting ways at the stations: holding question-answer sessions with health and agricultural experts located remotely, inviting people to comment about local problems, do live field reporting of elections and meetings, etc. Telephony also added considerable differentiation to GRINS as compared to other commercial automation systems; this was essential since by this time we had started talking to other players in the CR ecosystem and wanted to make a good case for them to partner with us.

Offline Telephony: Soon after online telephony deployment, we started getting other interesting requests too. A CR station wanted to run a quiz program where they would announce a question each day, and invite people to call and leave their answer. However, they could not assign a dedicated staff member to receive these calls. We therefore developed an offline telephony feature so that an automated prompt was presented to the caller to record their answer after a beep. This feature is now one of the most heavily used features in GRINS. Stations have used it to solicit grievances about government schemes which are later collated and broadcast on air, record requests for songs, seek feedback on specific radio programs, and even hold auditions for singing competitions!

Internet Streaming: We also added Internet streaming capabilities in GRINS based on requests from some CR stations. We heard woes of a few stations about problems they faced in getting advertising revenue. Advertisers wanted to be sure that their ads were broadcast at designated times but the signal did not reach them, and hence they were reluctant to give ads. Some stations had Internet access, and so we set up Internet streaming of their live broadcast. Advertisers could now be pointed to the website; this helped in a few cases where advertisers also had Internet access at their premises. The feature also brought other advantages to the stations, because they could now point potential funding agencies to their live streaming link as well. Seeing this, we have now also put up a web-page for each CR station running GRINS, at our own cost, so that stations can have a web presence to attract funding.

SMS: We are now building SMS support into GRINS since feature requests have come in from stations to conduct polls on SMS, send SMSes to their volunteers and community members, and improve their advertising revenue by including ads below their outgoing SMSes.

In conclusion, we feel the key learnings here are (a) to start with a few essential features likely to be used, (b) work hard to gain confidence of the users, and (c) be conservative about our own assumptions regarding feature needs, and instead go with user demand that seems reasonable. We next discuss the evolution of GRINS hardware and installer software over the past 2.5 years.

4. GRINS INSTALLATION

CR stations are quite varied in terms of feature demands, cost concerns, and even technical setup. We wanted to design GRINS to be as flexible as possible to cater to different demands, but it turned out to be hard. We are now inclined towards standardization for easier technical support than to-

wards flexibility. We next describe our insights for choice of hardware setup, and then discuss software installation.

4.1 Hardware Setup

Initial need for flexibility

GRINS runs on a Linux based computer, primarily because the open-source telephony engine we use is only available on Linux. During our pilot deployments, we however found stations to be uncomfortable with the idea of transitioning to a Linux operating system from Windows. We therefore designed GRINS with a service oriented architecture [7], where different tasks of GRINS are performed by services that can run on different physical machines and talk to each other using a simple inter-process communication mechanism. We then set up GRINS at the pilot stations in a two-machine configuration. The user interface was provided on a computer running Windows, and the rest of the activities including content and telephony management were carried out on a separate Linux computer.

The two machine set up was a compromise between the station staff's desire to work on a Windows based machine and the GRINS telephony component's dependency on Linux. The service oriented architecture of GRINS allowed such flexibility. However, as we discuss below, an increase in flexibility increased the possible points of failure and also increased the cost of maintainability.



Figure 5: State of cabling at Radio Bundelkhand. The device in the front in both the images is the Ethernet switch connecting the two GRINS computers. The device seen in the background is the Linux computer of GRINS.

Flexibility vs stability

Increased flexibility also increased the probability of failure because now two computers must run correctly and communicate correctly for GRINS to carry out its tasks. Our theoretical concern became a reality when the LAN connection between the two computers was regularly disrupted at Radio Bundelkhand which was a result of poor placement of cables and devices. As shown in Figure 5, movement of staff feet often caused the ethernet switch to lose power and reboot. Another challenge we faced was maintainability of GRINS across single machine and multiple machine installations: every code change needed to be designed for and tested against a large permutation of configurations created by different GRINS services running on different machines. This significantly increased our development time, and in-

creased chances of introducing bugs during upgrades.

Based on this experience, we now only provide a single Linux machine configuration for CR stations.² We have even converted the three pilots to single machine configurations. While stations initially complained about GRINS computer running on Linux, significant improvement in the usability of Linux distributions, frequent loss of data due to viruses in Windows systems, and reduced cost of deployment in a single machine setup have helped us convince the stations to use this common configuration. Using a standard setup at all CR stations also allowed us to provide more effective technical support as is outlined later in Section 6.

Cost vs stability

Our choice of hardware for GRINS installations was influenced by a careful study of the trade-offs between cost and stability. Our pilot deployments used an Analog Telephony Adapter (ATA) [1] to connect a PSTN line to GRINS computer. We chose to work with PSTN lines as against GSM connections since ATAs cost roughly half the price of a GSM gateway. However, none of the ATAs we tried could reliably collect caller-id information from the PSTN lines in India, which limited our ability to build a listener database in GRINS. In addition, many CR stations were remotely located and did not have a PSTN line available at the station premises. We chose to use a GSM gateway to overcome these limitations, even though it meant an increase in the cost of GRINS installations. The GSM gateway would also allow stations to send and receive SMS, a feature we intend to build into GRINS soon. Thus, we realized that even when building solutions for developing regions, cost is just one of the many factors that impact choices. In our case, practical constraints of using an ATA, and the stability of a GSM gateway, were more important than the cost of hardware.

4.2 Software Installation

As the GRINS hardware setup went through several stages of evolution, its software setup and configuration also evolved correspondingly. In addition, while most installations in India have been done by us, we have now also started guiding remote installations in Africa. We found that both required a different set of priorities.

Local installations

The first few GRINS installations were done by developers of the software. They manually copied binaries and made the software and hardware configuration settings at the station itself. Checklists were maintained to make sure each task was carried out correctly. Later we built installation scripts to somewhat automate the process.

As we scaled our deployments, we recruited technical support engineers to provide support to existing deployments and to carry out new installations. However, initially the engineers were not skilled enough to use the scripts or manually copy the binaries. The installer was therefore converted into an Ubuntu package [4], which allowed automatic installation of dependencies through the Debian package manager, and uninstallation and upgrades when required. With some training on Ubuntu package management, support engineers were now able to install GRINS with ease.

²We did not consider running Linux as a virtual machine on a Windows system due to stringent latency requirements in audio routing and the need to work with low end commodity hardware. [6]

Remote Installations

In spite of the package manager utilities, the installer was hard to use for a person with little knowledge about Ubuntu packages. We have had some interest from community radio stations in Africa to use GRINS, and realized that several organizations may not even have the necessary skills to use the package manager in Ubuntu. We have therefore now combined the GRINS installer with an Ubuntu installation disk. One now simply has to install Ubuntu on a computer, and GRINS is installed automatically with it.

Thus, when building solutions, careful thought to the installer utility and the skills of the people using it is necessary. Our experience shows that easy of installation can go a long way in increasing scale of deployments, particularly in the developing regions context.

5. TRAINING CR STATION STAFF

In this section, we present various strategies and trade-offs in training the CR station staff to use GRINS.

Repeated hands-on practice works

Table 1 shows the educational profile of 58 staff members working at 12 of the CR stations running GRINS. Although station managers and technical support staff had prior exposure to computerized systems and radio equipment, they were a minority at the stations. The other staff, who were responsible for carrying out all the GRINS related tasks, had little prior exposure to computers or radio stations. We found that repeated hands-on training is a suitable technique to train this demography.

Table 1: Profiles of 58 staff working at 12 of the CR stations using GRINS.

	Station Manager	Technical Support Staff	RJs and Other Staff
Total people interviewed	12	10	36
With Bachelor's degree	67%	80%	31%
High school or less	25%	20%	69%
With prior exposure to radio equipment	33%	40%	8%
With prior exposure to computers	67%	100%	25%

During our initial three pilots, each time we added a new feature or redid an earlier feature, we spent significant time of the order of almost a week in the field to train the staff in using the features. We followed a hands-on approach where we assigned specific tasks to one staff member while the others watched and prompted the person. We would then redo the exercise with other staff members until most of them picked up all the steps. This method of practice through guided repetition worked at all our three pilots for the program scheduling and live speech archival features of GRINS.

Utility of a feature drives its learning need

Our experience with the library widget was however different. After training on the second iteration with improved



Figure 6: Staff at Kumaon Vaani CR station (front) being given hands-on training on GRINS.

content tagging, staff at two stations remembered how to use the tool, but staff at one station seemed to forget it despite repeated trainings at gaps of two months or so. We found that here the station manager had not mandated the staff to tag content, unlike the other places. Thus, if the staff felt the need for a particular feature, being trained on it even once would suffice for them to learn the feature.

Remote training vs in-person interactions:

In our subsequent deployments, to be cost effective, we established the norm of spending only two days post installation for training. We then provided remote training over the phone or through user manuals. This has worked at 10 stations, but we get frequent queries from 6 other stations, mostly those having staff with little prior exposure to computers. For these stations, we found that explaining workflows such as content tagging which entails a step-by-step process, and telephony which requires an experimental setup for somebody to first place a call into GRINS, were hard to explain over the phone. User manuals were also not sufficient, probably because they were in English. As an alternative, we therefore thought of printing posters with flowcharts outlining the workflows graphically, which could be put up as a ready reference on the station walls. A section from one of our posters is shown in Figure 7. However, even then requests to visit the stations again for refresher training continued.

We are currently exploring the use of demonstration videos with step-by-step instructions on using each feature of GRINS. Till the videos are available we are offering a refresher training package at a minor charge to cover our travel costs, but so far nobody has opted for a paid re-training. Instead, informal arrangements are what work. If we are scheduled to make an installation visit somewhere, we let other nearby stations know if they need us to visit, and then claim local travel reimbursements. This of course means that we are not able to resolve queries at stations right when they arrive, but stations seem comfortable with a delayed resolution to feature usage clarifications.

6. TECHNICAL SUPPORT

Over the last 2.5 years of GRINS deployments, there were

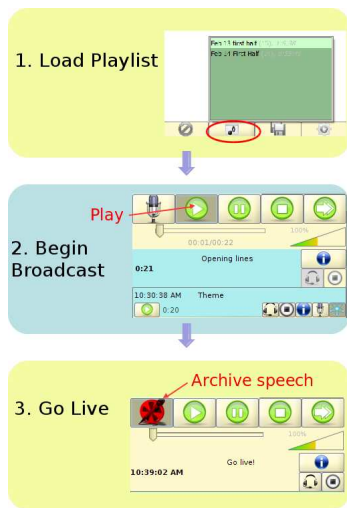


Figure 7: A section of the GRINS poster used as a ready-reference

several instances where the stations needed technical support. Initially software bugs were responsible for a large percentage of issues. However, as GRINS matured, faults related to misconfiguration and staff error required technical support most often. In this section, we present our experiences in providing technical support to CR stations based on data gathered between August 2010 and September 2011.

Moving parts fail often

Figure 8 shows the distribution of the faults identified upon requests of technical support from the stations. Out of a total of 45 requests for technical support, 11 or 25% were due to faults arising out of cables or mixer mis-configurations. Cables and mixers are the most accessible components in a radio station setup that can be tinkered with and are the hardest ones to remember on how to set them back. Figure 9 visually illustrates the messiness of mixer configurations and cabling at CR stations. It is easy for the staff to accidentally press a button or mis-configure the settings while cleaning equipment or experiment with the setup. Sometimes even when mis-configurations have nothing to do with GRINS, stations ask for our help because they consider us as mixer experts now.

Combined with mixer and cable faults, IP address changes and other mistakes form 47% of the total faults that were a result of staff error. This highlights the need to particularly focus on building tools to diagnose faults that are most likely to happen due to human error. Premi [8] has built a logical framework to identify basic cable faults in community radio stations. The framework takes as input the cable configuration of the station, and asks the staff a set of questions to automatically infer the exact point of failure. The diagnostics tool in GRINS however only examines software configuration errors, disk space availability, GSM gateway availability, etc. We intend to include the fault identification framework in the diagnostics tool, and enhance it by building different mixer models into it. We believe this change in GRINS will significantly reduce our technical support load.

Rural power supply is poor

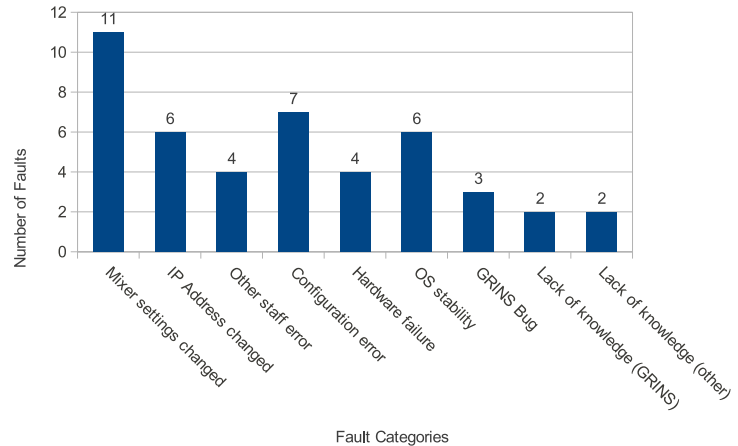


Figure 8: Faults identified, based on technical support requests between August 2010 and September 2011



Figure 9: Complexity of mixer settings and cabling at Radio Bundelkhand

Figure 8 brings out another interesting fact: a large percentage of CR stations routinely experience electrical fluctuations and outages. The four hardware faults caused by disk crash and motherboard failure seem to be results of very low or high voltage levels in the electrical supply. Similarly, 6 OS stability faults were file system corruptions caused by electrical outage leading to improper machine shutdown. Together hardware failures and OS stability faults form 22% of total faults, even when all the stations use UPS and stabilizers to regulate the electricity. Our observation corroborates experiences of Surana et al. [10, 9] in building wireless network infrastructure for rural regions.

Providing technical support only over phone is hard

In addition to the GRINS diagnostics tool, we used communication over phone, remote machine access over the Internet, and physical visits to provide technical support to the CR stations. Table 2 lists how each of these modes were used in identifying and resolving faults between August 2010 and September 2011. We can see that 41 out of 45 faults were identified over the phone, but additional tools were required to resolve 29 of them. Our experience indicates that phone calls help diagnose faults, but are alone insufficient to rectify

them.

Resolving faults over the phone has its limitations due to the lack of necessary skills at the CR stations. Station staff are unfamiliar with the Linux operating system, they are usually uncomfortable with English, and have rarely used non-alphanumeric characters found on a typical keyboard. This makes it hard to fix faults that may require even minor computer skills at the other end. To elucidate the challenges, we describe examples of two tasks that we tried to instruct the station staff to execute over the phone.

- **Open command line terminal:** Even simple things like going to the “Applications” menu present on the top left of the screen, then going to the “Accessories” sub-menu, and then clicking on “Terminal” were hard to get done. It seems that the English words were hard for the station staff to read³. Often the staff would even give up reading the words and say that there is no “Applications” menu visible on the screen.
- **Redirect the list of files in the current directory to a file:** This is achieved by typing `ls -l >list.txt` in the terminal. First, we had to use the word “minus” instead of “hyphen” to describe the “-” in the command. Second, we also had a difficulty in getting the staff to type “>”, since typing this character requires pressing “SHIFT + >”. The station staff did not know where the “>” key was, and differences in keyboard layouts across stations made it harder for us to guide them. Further, typing `ls -l >list.txt` on the terminal does not produce any output but silently writes to the file. The staff would report back to us saying that “nothing happened” and the only way for us to find out if the command execution was successful was to make them check if the file was created.

As a result, as shown in Table 2, more than 63% of faults could not be resolved over phone calls, confirming the challenges highlighted above.

Table 2: Modes of Communication for Technical Support. How each mode helped in identification and resolution of faults between August 2010 and September 2011

Mode of Communication	Faults Identified	Faults Resolved
Phone Calls	41	16
Remote Desktop	0	9
Remote Desktop and Phone	3	6
Reverse SSH	1	5
Reverse SSH and Phone	0	2
Physical Visit	0	6

³Audacity, an audio editing tool, has similar menus, but the staff are adept at using it. So it seems that the staff have learned Audacity through repeated practice without necessarily understanding the concept of menus. This further validates our strategy of repeated hands-on practice for GRINS training highlighted in Section 5.

Remote Internet access is ideal: To overcome the challenges of resolving faults over the phone, we try to use Internet based tools wherever possible. Internet is available at about 70% of our deployments. Depending on the speed of Internet connectivity available we either use TeamViewer [3] (a remote desktop tool) or reverse SSH [2] to connect to the GRINS computer. Both these tools allow us to connect to the GRINS computer even when the computer is behind a proxy or a NAT box. As Table 2 shows, 4 faults required Internet tools for identification and 22 or 50% of the faults required Internet tools for resolution.

The faults that required Internet tools for identification were typically those which we did not expect to occur, and hence did not check for while on the phone. For example, in one case the soundcard configuration had been changed in Linux, which was unexpected because the staff did not seem to have the necessary skills to do so. Similarly, in another case the output volume setting for GRINS had been set to 0, which was unexpected because we believed that the staff were aware of the setting and would have made sure that it is correctly set.

IP address changes and installation/reconfiguration errors, which formed about 31% of faults, were resolved over the Internet alone. Another 19% faults were resolved using a combination of Internet tools and phone calls; these typically required executing commands on the GRINS computer as well as adjusting cables/mixer settings.

When a fault cannot be resolved over a phone call or using Internet based tools, we have to physically visit the CR station. There have been 6 such instances between August 2010 and September 2011. However, four of these six visits were taken up because they coincided with hardware upgrades and meetings with station management. Thus, in only two cases or 4% of the faults, we had to visit the station exclusively to resolve the problem. Even here, we noticed that physical visits were required because the station staff were unwilling to follow instructions on the phone. In one case, the file system on the station’s GRINS computer had crashed and the station staff was not comfortable even reading the information appearing on the screen; while in the second case, bad sectors had appeared on the hard disk and the staff was unwilling to run special tools for data recovery.

Thus, physical visits are rarely required and most faults can be fixed through a combination of phone calls and remote Internet access. That said, there is clearly lots of scope to redesign OS instructions, hide details, enable automatic recovery, etc, for low-skilled users.

Standardization of installations helps provide effective support:

Stations often have very different cable and mixer settings. To enable us to provide support in such varied situations, we reset their settings and attempt to keep the hardware, cabling, and configuration common across CR stations. For example, irrespective of the mixer model, broadcast audio from the GRINS computer is always fed to the mixer on input line number 4. In cases where such standard cabling cannot be done, we take photographs of the mixer settings and the cabling to refer to it when providing support. Similarly, whenever possible we use the same computer hardware, GSM gateway, and soundcard for GRINS installations. We also try to keep the IP address of the GRINS computer and the GSM gateway same across CR stations. Such “stan-

standardization” allows us to provide support more easily than context switch between different settings.

7. SALES

CR stations do not have the necessary technical skills to install GRINS and integrate it into their station setup. We therefore install GRINS with relevant hardware and train the station staff on its usage for a charge. Thus, even though GRINS is a free software, its costs of hardware, installation and training for a CR station are non-zero, and we have to “sell” GRINS to CR stations. Table 3 lists the approximate cost of GRINS installation for a CR station. Even at a nominal cost of USD 2500, significant effort is required to sell GRINS as most NGOs are short on funds to invest in technical systems that improve quality and efficiency. In this section, we describe our experience of selling GRINS to NGOs, and identify interesting purchase practices that may be useful for other ventures operating in this space.

Table 3: Cost of GRINS installation as borne by the CR stations.

Item	Cost
Computer	USD 1000
Soundcard	USD 200
GSM Gateway	USD 450
Installation	USD 250
Training	USD 300
Cables and Ethernet Switch	USD 150
1 Yr Remote Technical Support	USD 150
Travel and accommodation costs	At actuals
Total	USD 2500

Ecosystem and market segmentation

The community radio sector in India is very small with all the players strongly connected to each other. There is an interesting mix of several forces influencing the movement: idealism, strong egos, opportunity to profit in a growing sector, need for sustainability; and they often conflict with creating impact. The CR ecosystem comprises of the following entities:

- *Parent organizations of CR stations:* The Indian CR policy mandates that stations cannot be independent entities, but have to be owned and operated by non-profit organizations. Over the years, two distinct segments have emerged: a higher-end segment consisting of mostly large NGOs and educational institutions, and a lower-end segment consisting mostly of small NGOs. The higher-end segment typically use expensive station setups, while the lower-end segment run bare-bones stations that suffice for community building activities.
- *FM transmitter vendors:* There are three authorized vendors in India, out of which only one is an indigenous manufacturer and hence has lower prices.
- *Training organizations:* These organizations train the station staff in community engagement, basic journalism practices, and audio production. We estimate the

number of such organizations to be between six and eight in India.

- *System integrators:* They offer an end-to-end solution to radio stations, helping them identify the right equipment, set up the studio, put in purchase orders, and give basic technical training. There are about four popular system integrators, all mostly catering to the higher-end segment that can afford their services. Small NGOs instead often take the help of training organizations to identify the set of technologies suitable for their needs, and then connect with vendors directly.
- *Radio automation system providers:* This includes us in the lower-price category, plus a few other commercial closed-source systems that partner with system integrators and target the higher-end segment. The feature-set offered by these commercial systems are an overload over those required in typical community radio settings; CR requires feature-sets such as low-cost telephony for community engagement, being able to maintain community databases, backup content, etc, and not advertisement management or automatic playlist construction to queue music, which are unique selling propositions of the commercial systems. No commercial system to the best of our knowledge provides as low-cost telephony integration through GSM gateways as GRINS; a few systems provide analog taps and other phone interconnection methods that are 10x more expensive [6].
- *Advisory forums:* The community radio forum of India and a few other non-profit foundations organize workshops to create awareness about community radio amongst NGOs, and guide them in their license application process.

Aligning with idealism: Profit vs impact

With less than 150 CR stations, the size of the CR market in India is fairly small. But there is significant differentiation within the market segments of small NGOs and educational institutions respectively – small NGOs rely on training organizations who are in turn preferentially networked with a set of low-cost vendors for FM transmitters and other equipment, while educational institutions and large NGOs outsource their station setup to system integrators who are then networked with comparatively more expensive alternatives. The small NGO space also has a dominant sense of idealism because most players have been historically associated with pushing the community radio legislation in India and opening up the sector. Hence, there is a general feeling of ownership and a normative definition of how a CR station should operate. The unstated objective here is to ensure that the station is community driven, irrespective of the technical setup; and choices are inclined towards low cost setups, because small NGOs cannot afford extravagant installations. On the other hand, educational institutions often see their stations as training labs for their students, and large NGOs tend to view their stations as another “project”. Thus, choosing to work with small NGOs would have meant greater chances of impact but lower profits, and choosing the higher-end segment would have meant vice-versa. We consciously chose to establish ourselves in the market segment

of small NGOs since they would gain most from the low-cost setup of GRINS, and thus priced GRINS at level that was affordable for this segment. However, it made GRINS less attractive for system integrators of the the large NGO segment as there was very little profit to be made.

Making our presence felt

As a first step towards sales we needed to let the players in the segment know about our product. We created awareness about GRINS by attending workshops and facilitating training sessions organized by different advisory forums. Our presence at these workshops helped put GRINS on the radar of many organizations.

Acquiring goodwill and credibility

We wanted to let various players in the lower-end segment know about our genuine desire to work with them. We developed relationships with training organizations and low-cost transmitter vendors, so that they would promote GRINS when they made purchase suggestions to small NGOs. We now have good relationships with most training organizations, and 6 of our deployments have actually happened through a partnership with the low-cost FM transmitter manufacturer in India. Due to the small market size and strong connectedness between CR stations, word-of-mouth recommendations traveling across NGOs have also helped; 4 of our deployments happened through references from existing customers.

Selling to NGOs: How and when

We observed that most NGOs set up stations with funds from grant agencies that budget according to a certain capital cost and some operational costs. Since several stations to whom we tried to make direct sales were already up and running, justifying the cost of an automation system as an operational expense was hard for these stations. Audit practices in the development sector are quite strict, and therefore the station would have to re-budget purchases with their donors which can take anywhere between 6 months to a year. We used a number of ways to make purchases easier for these stations, such as reducing costs by clubbing together travel to nearby stations, installing GRINS on their unused hardware rather than buying new hardware, accepting payment through installments, and we even made pro bono installations in cases where the station was doing phenomenal community service but was unable to pay.

Interestingly, the same type of NGOs with similar financial standing were however comfortable in making full purchases at the time of setting up of a new station. We have therefore realized that the best case scenario is to network with stations while they are putting together their budget, because at that stage NGOs (and probably even donors) seem to have a free hand! Thus, the ease and success of making a sale to a CR station depends heavily on the stage of establishment of the station.

Current financial status

Catering primarily to the lower-end segment clearly meant that we had to price our solutions low (unless we adopted some complex differential pricing policy). In spite of this, our cost coverage is almost break-even now to cover the salaries of a technical support engineer, but without including our development costs and that of pro-bono installations. Sustainability is therefore challenging, but there are talks

among the Indian bureaucratic circles of creating a community radio support fund to help small NGOs and technical organizations, that we may benefit from.

8. CONCLUSION

In this paper, we presented our experiences of designing and commercializing GRINS in India. We showed that our assumptions about the feature needs of CR stations led us to spend significant time and effort in building a library feature that is used infrequently, and features such as telephony that were built on-demand are frequently used. We highlighted the need to prioritize stability of hardware over cost when scaling pilots, which is contrary to the norm often discussed in the developing regions context. We also showed how standardization of deployments helps reduce technical support costs and even makes support provisioning more effective. Our analysis of the faults reported by CR stations over a period of one year has shown that most faults are a result of mis-configurations due to staff error, and that building diagnostic tools for such faults can go a long way in reducing support load. Our analysis also showed that remote Internet access combined with phone is sufficient to resolve most problems, with physical visits required only when the staff are not able to follow instructions over the phone. Finally, we showed that the most effective strategy for selling to NGOs is to combine the sale as a part of a larger project. We believe these insights will be relevant for the technology for development community, particularly those looking to scale technology solutions in developing regions.

9. REFERENCES

- [1] Analog telephony adapter. http://en.wikipedia.org/wiki/Analog_telephony_adapter.
- [2] Reverse SSH. http://en.gentoo-wiki.com/wiki/Reverse_Tunneling.
- [3] TeamViewer. <http://www.teamviewer.com>.
- [4] Ubuntu Packaging Guide. <https://wiki.ubuntu.com/PackagingGuide/Complete>.
- [5] M. A. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *Workshop on Computer Interaction and Information Retrieval (HCIR)*, October 2008.
- [6] Z. Koradia, A. Premi, Balachandran C., and A. Seth. Using icts to meet the operational needs of community radio stations in india. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, pages 21:1–21:9, New York, NY, USA, 2010. ACM.
- [7] Q. Mahmoud. *Middleware for Communications*. John Wiley and Sons Ltd., 2004.
- [8] A. Premi. Framework for Inferring and Debugging Community Radio Station Configurations. <http://www.cse.iitd.ernet.in/aseth/act4d/reports/grins-formal-expression.pdf>.
- [9] S. Surana, R. Patra, S. Nedevschi, and E. Brewer. Deploying a rural wireless telemedicine system: Experiences in sustainability. *Computer*, 41(6):48–56, June 2008.
- [10] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer. Beyond Pilots: Keeping Rural Wireless Networks Alive. In *USENIX NSDI*, 2008.