Revitalising the Rebati project

G. Mohanty

November 20, 2007

1 Introduction

Three, and a half years ago, we had started the Rebati project (`http://oriya.sarovar.org`) with a great deal of enthusiasm, and fanfare. Apart from some limited successes in furthering Oriya support, and in localisation of FOSS applications, we have failed miserably in our main goal of getting people to use Oriya on Free/Open Source Software (FOSS) platforms. My intention is not to point fingers, and I count myself as being among the people responsible for the lack of success. What I would like to do now is make an honest appraisal of the causes of the failure, and try to find remedies.

While I will name some specific goals and approaches in this document, please note that these are only proposals from my immediate personal thinking, and it is important that they derive instead from a consensus among the community.

2 A fresh approach

It is time to leave behind ideas that have not worked. Here are what I see as the essential ideas around which to build a revitalised project:

- *Build on a core vision, rather than on ad-hoc plans:* The crux of the problem lies in the fact that we have managed to lose the excitement, and the motivation that comes from first seeing Oriya actually work on Linux applications. Instead, we were overly obsessed with planning, and the more our plans did not work, the more we became convinced that a better plan, a more rigorous process, would save us. A similar problem is also at the heart of the current stagnation in the IndLinux project (`http://indlinux.org`).

- *Value people over processes:* Rather than emphasising processes, such as a particular set of rules that must be followed for accepting a translated file, and asking that newcomers jump through certain hoops, we should focus on attracting dedicated people to the organisation, and giving them added responsibilities, with a free hand to achieve these.

- *Building in adaptability:* Technological development, poses a set of ever-changing challenges, and like in biological evolution, the most successful projects are the ones that can meet these with an adaptable set of responses. This has two aspects to it, the first being to be quick to recognise when things are not working, and the second being responding rapidly, and without pre-conceived notions, to new problems that will inevitably arise. Thus, we should seek to embrace change, instead of shying away from it.

By contrast, traditional project planning starts with a fixed set of approaches to a fixed set of goals. There is usually little room for managing change, and at best it can achieve only what was envisaged at the beginning.

To make these ideas tangible in the context of the Rebati project, here is what we might come up with:

- The core vision should pretty obviously be ensuring the spread of Oriya usage on FOSS platforms, namely Linux at the moment. This broad vision will need to be broken down into smaller goals, e.g., which Linux distribution and what applications to focus on, how to provide a migration path to users from proprietary applications, and operating systems, etc.
- The first step should be the building of a community. While we have said this before, I now believe that this is so fundamental that there is little point in trying to do anything else without having achieved this. So, this should be the immediate focus of all our efforts. This community should have room for both local people in Orissa, and tap into the vast Oriya diaspora in other parts of India, and also abroad. We should ask on a commitment from people. This can be however little time that they can spare, but we should try to hold them to it, a small commitment that is actually seen through being much more worthwhile than larger promises that never materialise.
- We need to learn how to break up broader goals into smaller tasks, and approach these in a time-bound, iterative manner. I.e., we set time limits to each task, and do our best to hold to these, and that we approach our goals in a manner that constantly improves on past work. Thus, for achieving any particular goal, the first step should be a crude one that meets broad objectives, with later steps filling in details, and adding refinements. The idea behind this is that we have something to show immediately, and can later balance the available time against the degree of perfection to be sought in achieving the goal.

3 Concrete plans for the next year

Here are the concrete steps that I believe should be initiated immediately, with a time line of a year to achieve things. I have consciously stuck to broad areas, as it is the vision that is important, and the details will emerge from a participatory planning process:

- *Support for Oriya on FOSS platforms:* This involves making sure that Oriya works seamlessly for the end-user on the chosen deployment platforms, including inputting, editing, rendering, printing, saving, and re-editing.

To a large extent, this is already possible, but we need to decide on a specific set of items to target. To my mind, these should be:

  - The Utkal Oriya font: This has recently been worked on by Rahul Bhalerao, who has fixed many issues with the rules, thus allowing it to work on the newest Pango renderer.
  - The SCIM input method: This is the best alternative at the moment, though still far from perfect. Its advantages are that it nowadays comes bundled with most distributions, and allows a variety of keymaps, including those for old-style 8-bit fonts, under one common interface.
  - Rendering: The good news is that major applications like GNOME, KDE, Open Office, Scribus, etc., are heading towards a common rendering back-end, which will make it easier to find and fix bugs. I plan on putting in work on the Indian language component of this, hopefully with help from some developers in Redhat India.
  - Specific distributions, and applications to target: Here is my list, and I propose that we not expand this beyond at most 8 areas, with 2–3 applications in each area:
    - Distributions: Ubuntu 7.10, which was just released, and, maybe Fedora 8, though I have little personal experience with the latter.
    - Desktop: GNOME, with XFCE, or KDE as possible alternatives.
    - Editors: gedit, or any basic editor on the chosen desktop. Most of these should support Oriya well.
    - Word-processors: Open Office.
    - Desktop publishing: Scribus.

- *Usability of Oriya on FOSS platforms:* Besides basic support, the use of Oriya should become painless to the end-user, rather than requiring her to download, and configure a bunch of packages. Making it as easy to use Oriya on a desktop as English currently is involves: (a) A minimal, and sufficient set of packages installed by default. This will require making a meta-package for the distribution of choice. (b) Translation of documentation: This is becoming more important than translating the user interface, as a fair amount of that is already done, and as most current computer users needing Oriya will already have at least some knowledge of English. (c) A live discussion forum that can answer issues that arise, and is actively monitored by Rebati team members.

- *Localisation, and technical terminology in Oriya:* Continued localisation work, i.e., the translation of the user interface of the application into Oriya should be a continued objective, but we should also look at other aspects, including:

  - Technical terminology in Oriya.
  - Dictionaries, for spell-checking, and other purposes.
  - Language corpora for a variety of applications, including OCR.

  - Spoken Oriya corpora, for use in speech-to-text.

- *Deployment, and maintenance support:* To my mind, it is beyond our foreseeable capabilities to provide support for migrating organisations to Oriya on Linux, and in providing them sufficient support to be able to manage their machines. However, if we build a base that works well enough, and is available as open-source, it then becomes worthwhile for a commercial organisation to provide such support. For example, Redhat is in the process of deploying some 10,000 Linux machines, with a Bengali interface, for the West Bengal government, and it should be possible for a small local company to start providing paid support in Bhubaneswar.

  What we can do as part of the Rebati community is to run a showcase Oriya FOSS project in one or two places, say at Srujanika to start with.

4 Resources required

The number of people required to run such a revamped project successfully is hard to estimate without first finalising the list of tasks to be taken up. A ball-park estimate would be one full-time person in Bhubaneswar, and one full-time person elsewhere, but available online. Others can fit in according to what time they have available, and the task list can be expanded if we get a large response.

Sarai, CSDS (`http://www.sarai.net`), can currently provide 1–2 fellowships for work in this area. As Sarai itself is in a state of flux, and as there are many demands for the limited amount of funds that we have available, continued support cannot be guaranteed, but such should be possible, at least until such time that the project can become self-funding.