

Sahana Internationalisation Languages and beyond

Dominic König, RN
March 24, 2009

1. Introduction

When people discuss making Sahana applicable worldwide, the first topic that is usually raised is the translation of text elements of the user interface into the user's native language.

As other software applications, Sahana uses runtime functions that perform internal dictionary lookups and replace text elements in the user interface by their corresponding translations before displaying them to the user.

At first glance, it would appear that there would be nothing else to do other than to translate these internal dictionaries for all the required languages and to install and configure them at the Sahana server.

Well, we all know that it's not that easy...

2. Internationalisation and Localisation

Sahana is intended to be a general-purpose disaster management software application for worldwide use. As a result of this global demand, there is significant work underway adapting it to many different types of disasters and their corresponding typical issues in preparedness and response.

On the other hand, this also requires that Sahana is adapted to various languages and regional differences. In some occasions a single Sahana server may have to meet the needs of different regions at once.

This may include not only different languages and scripts, but also different cultural specifics, such as names and titles, social and administrative structures, measurements and currencies, calendars, government assigned numbers and so on.

In principle, the adaption of a software system to regional specifics has two aspects:

- (1) To design the software in a way that allows adaption to different languages and regional differences without having to rework the code. The respective measures are usually summarized as Internationalisation (i18n).
- (2) To adapt the software to a particular language or region - also known as Localisation (L10N).

For Sahana, as for many other open source projects, the internationalisation aspect also includes the interconnection and integration of a diverse global community of researchers, developers, users and other interest groups.

3. Localisation Issues

3.1. Language

The most significant localisation issue is the translation of the user interface into other languages than the one originally used.

It is common practice in software systems is to place any text of the user interface in resource strings which are loaded during program execution as needed. These strings are usually stored in resource files, which are relatively easy to translate.

Sahana uses the *GNU/gettext* library to load the appropriate resource strings at runtime. *GNU/gettext* is very popular and common in open source projects, hence there exists a large number of tools to maintain the resource files (so called PO files) - for example, the *Translate Toolkit*, a set of python scripts for almost all imaginable use-cases of resource file maintenance, and *Pootle*, a server software for web-based collaborative translation which is based on the *Translate Toolkit*.

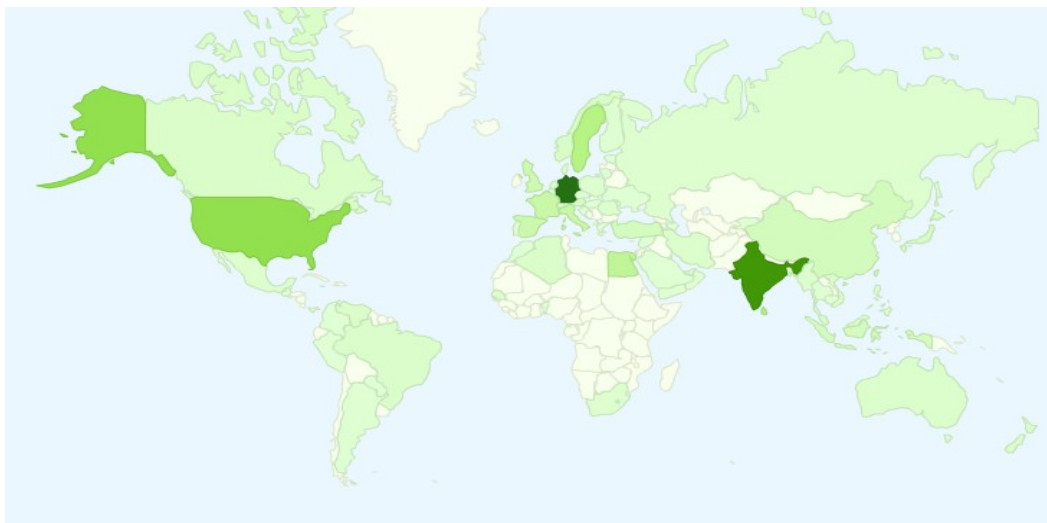
Since 2008, the Sahana project has operated its own *Pootle* site as a centralised collaborative platform for user interface translations. This platform is widely accepted by FOSS translators, and selection of *Pootle* has already resulted in a number of new translations. On the other hand, notwithstanding the advanced tools and collaboration options, it remains a time-consuming issue to involve new translators, especially those that lack the required IT skills.

While translating existing text to other languages may seem easy, it is more difficult to maintain the parallel versions of texts throughout the lifecycle of the software - for example, if an interface message is modified, all of the translated versions must be changed as well. This

might cause serious delays in the software release process,. Unless there is a strong translation community to translate changes at short notice, the Sahana users suffer until the corrected translations have been made.

To counter the performance bottlenecks, major efforts are being undertaken to build project capacity in, and promote, permanent well-prepared and highly available localisation teams around the world.

These teams represent one of the largest parts of the contributing community in the Sahana project and need corresponding co-ordination, integration, recognition and support.



*Sahana Online Translation - Worldwide Distribution of Contributors
(Source: Google Analytics, translate.sahana.lk)*

3.2. Character encoding

To support various the scripts from a wide variety of languages, Sahana consistently uses UTF-8 character encoding for both data exchange and user interface. UTF-8 encoding is natively supported by the most platforms and client applications, especially open-source.

Occasionally, some environments require additional software to be installed and configured to enable consistent UTF-8 encoding. Additionally, specific fonts may need to be installed. These issues may seem minor, but they are nevertheless an important aspect of localisation. If necessary and available, Sahana can provide add-on packages as well as documentation for corresponding solutions.

3.3. Writing direction and other conventions

Some localisation issues may require more profound changes such as code addition or modification.

For example, the writing direction determines the overall orientation of

the user interface, and requires customisation of stylesheets and certain interactive widgets.

This approach also applies to date/time formats, number formats, plural forms, capitalisation, punctuation, and other writing conventions. Most of these are customized by the configuration settings in the underlying platform.

Sahana already supports highly customizable styling of its user interface by means of exchangeable style sets ("themes"), and the basic interface structure is designed with flexibility in mind. A set of theme templates for Sahana with different writing directions is currently still under development.

In the same context, it is also important to consider appropriateness and comprehensibility of non-text user interface elements, such as colors, images and symbols, for the particular region. In SAHANA, this can as well be easily adapted through theme options as needed.

3.4. Data Model

Certain elements of the underlying data models, for example personal data, also require some degree of localisation.

Names, addresses, and even government assigned identification numbers must be consistently readable and searchable even when switching to another locale, and identification numbers must be unique across all contexts.

On the other hand, conventions and tradition in names, family structures and social relationships differ considerably from region to region, and occasionally even between different ethnical groups in the same region, not to mention different languages and language scripts.

It is obvious that Sahana needs to provide solutions for the localisation of its data model for different regions as well as for consistent data handling in multi-ethnic and multilingual environments.

A partial solution is provided by Sahana as customizable option fields.

3.5. Regulatory Compliance

Storage, processing, and usage of data in Sahana, especially of personal data, is usually subject to certain legal and administrative regulations and restrictions. The emphasis here is usually on management of private information, data protection and integrity, freedom of information and other such aspects.

Sahana provides a very flexible model for authentication, authorisation and auditing (AAA). However, country-specific regulations may require different or additional functionality that go beyond the Sahana AAA model.

This applies as well for the right of persons to get knowledge and control, to certain extent, on that information that is stored about them and the particular purposes for which it is processed. Additional functionality may be needed to provide all the required capabilities, however this will be difficult as there is great global variety in privacy and data management legislation worldwide.

Another aspect of this is the implementation of certain standards in Sahana - mostly for interoperability reasons. Which standards are required may vary from region to region, depending on pre-existing solutions and infrastructure. Major international standards will likely be taken into account during application design and development, whilst some local standards may be object for localisation.

3.7. Terminology

Independent of the language actually used, it is mission-critical in Sahana for users and developers to speak a "common language", that is, to avoid ambiguous terms and fuzzy phrases. Wherever possible, expressions should be used in a way that they are clearly defined and understood.

Disaster response usually involves diverse competences, where every subject area has its own specific terminology. This complicates the localisation of Sahana, and involves a serious risk for translation errors, misunderstandings and thus limited usability and unsafe application.

To resolve this, the Sahana localisation team is working on different approaches:

- (1) Ontologies of terms in respect of their different contexts. A good template for such domain ontologies could be the International Classification of Nursing Practice (ICNP®). Such ontologies can be used by developers for data modelling and user interface design, and can also be helpful resources for translation.
- (2) Glossary of Sahana terms, primarily meant to support translators. If the glossary is translated first, then it can be used to assure consistency of translations by helping to define the context the term is used in.

- (3) Styleguide for user interface messages, which contains recommendations on how to avoid misunderstandings, inappropriate forms of expression and common localisation problems such as overflowing text boxes etc.

3.8. Maintenance of Localisation

As stated before, it is a major challenge to maintain localized versions throughout the lifecycle of a software product.

Simple release maintenance measures such as bug fixes can require extensive effort to propagate them to all localized versions, especially if the changes involve text or other interface elements.

To avoid serious delays on updates, it is necessary to have well-prepared and highly available teams for all localized versions, who can rapidly adopt changes.

This in turn requires reasonable overview and control on all existing localisation projects as well as efficient communication channels to notify relevant changes and request or provide appropriate support. Furthermore, it is good practice to include localisation teams in the software development process to ensure internationalisation and localisation are taken into consideration, before modifying the source code.

Translation of documentation is similar and involves the same maintenance issues as the translation of the user interface. Keeping multilingual documentation up-to-date, however, currently exceeds the available resources of the Sahana project as documentation has significantly greater volume of text to be translated. However, this has to be seriously considered for the future - including a suitable revision control system for the documents and corresponding translations. One approach may be to utilise wiki software such as Mediawiki that supports the creation and modification of pages in multiple languages.

3.9. Integrating a global Sahana community

Internationalisation of Sahana is not restricted to the software or user documentation. Integration of a global community needs localisation as well.

That does not mean that every element of the Sahana project needs to be provided in multiple languages. It rather means, that support in a local language - for both users as well as contributors - can be much more efficient.

Moreover, experiences have shown that local events with discussions in local languages usually lead to much more agile and productive exchange of ideas, especially with and among people with less developed English communication skills. Localisation of certain portions of the developer documentation can lower barriers-of-entry for new contributors and improve the overall quality of collaboration - not to mention create a more inclusive and open environment.

"Localized" public relations may be somewhat essential and perhaps the most promising way to increase the appeal of Sahana.

4. Current focus

4.1. Translation

During 2008, there was a notable progress in the translation of the Sahana user interface, and Sahana is now currently available in 12 languages including (see table below). Broadly, this represents 40-50% of the world's population. The Sahana translation effort has attempted to target and prioritize specific languages based upon the speaking population of the major language.

Ranking (languages worldwide)	Language	Total Speakers (millions)	Native Speakers (millions)
#1	Chinese	1150	870
#2	Hindi	650	370
#3	Spanish	417	320
#4	English	1500	310
#5	Arabic	422	206
#6	Portuguese	235	180
#8	Russian	275	145
#10	German	180	100
#19	Tamil	80	68
#37	Burmese	42	22
#50	Indonesian	160	20
#63	Sinhalese	15	13
TOTAL		up to 3500	2624

Sahana: Translations Available

More translation projects, as well as refinement of existing translations, are being worked on. The Pootle server acts as the central coordination and collaboration platform.

All translations are published as separate language packs, which can be

installed on any Sahana server with minimum effort. The language packs for the latest stable release get updated at least every month.

Current languages that are a high priority to complete include French, Italian and Indian languages, but other translations need review or completion as well. Our aim is to reach 60-70% population coverage by major language by the end of 2009.

Ranking (languages worldwide)	Language	Total Speakers (millions)	Native Speakers (millions)
#7	Bengali	200	170
#11	Punjabi	105	88
#12	French	360	78
#20	Italian	120	60
TOTAL		up to 700	396

Sahana: Current Top Priority Translation Projects

4.2. Sahana Localisation Knowledge Base

All sorts of useful knowledge regarding the localisation of Sahana both in general and regional specifics is currently being aggregated into a well-structured knowledge base.

This knowledge base is aimed to support localisation teams and provide recommendations for developers in order to speed up localisation and improve quality. It is focussed on Sahana 1.0, and shall be publicly available at the same time as the version 1.0 release of Sahana.

Some portions of the knowledge base, for example a style guide for english text in the user interface, the Sahana glossary of terms and a detailed guide on how to translate Sahana are already available on the Sahana Wiki.

4.3. Other works

- Theme template for Right-To-Left orientation
- Support for Translators through automatic translation tools and use of translation pools (e.g., Launchpad)
- Localisation Add-ons/Packages like themes, language packs, pre-customized modules and region-specific add-ons
- Promotion of Local Sahana Teams

Sahana Internationalisation

March 24, 2009

5. Resources, Links

Sahana Project:

- Project Homepage: <http://www.sahana.lk>

Sahana Localization:

- Pootle Site: <http://translate.sahana.lk>
- Sahana L10N GoogleGroup: <http://groups.google.com/group/sahana-localization>
- Sahana L10N Blogsite: <http://backus.nursix.org>
- Sahana L10N Wiki: <http://wiki.sahana.lk/doku.php?id=translate:home>

Sahana Language Packs:

- Sahana Language Packs: <http://pub.nursix.org/sahana/locale>

Technical Resources:

- Translate Toolkit and Pootle: <http://translate.sourceforge.net>
- GNU/gettext: <http://www.gnu.org/software/gettext>

Other:

- http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers