18 . Representation Scheme for User Environment

Ms. Alka Irani, Senior Research Scientist, CDAC, E-mail : alka@ncst.ernet.in

Abstract:

Software programs are agents that communicate with humans using interfaces. User interfaces act as gateways to these agents and thus are responsible for the success and acceptability of the software.Purpose of this presentation is to stress the need for building a foundation for the communication medium using which these software agents can interact with humans in a way that makes localisation easy and effective.

Research in language technology, has reached a point where computers can easily input/output and store text in local languages as well as communicate in local languages. It is also clear to us that IT in local languages can make the benefits of IT reach all sections of the society rich and poor , urban and rural.

The technology has also made it comparatively easy to find cheaper and affordable solutions for the sectors like education and e-governance. Is the medium ready for dwelling in local languages? Can we localise the user environment quickly and deploy IT in non-IT sectors?

In order to get answers to these questions, let us first understand what is meant by the phrase 'Localising User Environment' by understanding each word.

In general, Environment consists of various systems, organisations and agents interacting with each other. User environment here assumes that the agents that provide services are computers and those who use these services are human beings. In order to get these services done by these agents called computers, users have to communicate with them through what is called user interfaces. Different types of user interfaces exist each having its own merits and de-merits. They are menu-based, form filling, command languages or query languages. They contribute to the 'user environment'.

Localisation in a very loose sense means changing these dialogues, interfaces to suit users from different backgrounds, different countries, from different domains and speaking different languages, coming from different cultures. Efforts were going on in localisation for last 10-15 years and it has become a separate branch of computer science discipline. IT Technology is changing so fast that yesteryear's solution becomes absolute this year. Localisation saves lot of efforts. We do not have to carry the burden of the legacy systems. We can inherit wealth of knowledge and software generated in other languages.

Internationalisation is the discipline that insists that the core program logic should be separated from the code for interaction component.

The Indix project undertaken at C-DAC, Mumbai is an effort to localise GNU/Linux Operating system at the core level. It is a project funded by the Ministry of Communications and IT under the TDIL programme. The immediate target of the project is to localise suitable components within the GNU/Linux operating system to enable applications to create, edit, view and print contents in 12 major Indian languages using both UNICODE and ISCII encoding. The basic philosophy behind Indix effort is on Commonly available computers Indian languages should be handled with same ease as English.

Primary contribution of this project is the development of a robust shaping engine for rendering a linear text stream to the graphics display with support for complex text layouts. This text-rendering pipeline involves syllable identification, context sensitive reordering of character codes, repositioning and mapping of glyphs, and rendering of glyphs on the screen (display) or paper (print). As the modifications were done at the core part of the Linux windowing system (X window system), many internationalized applications in Linux can work with Indian languages without re-compilation or modifications.

The project has implemented script shaping architecture, text handling libraries and Indic script handling using Open Type font for 12 languages so far and will migrate these components into the open source software Linux and X11(XFree86).

Indix can be thought of as just a step forward towards a solution for localisation of computing environment for Indian languages. Efforts are also going on to localise desktops in various Indian languages by various organisations and various voluntary groups like IndLinux. This essentially involves translating the menus translating the messages taking care of data formats, currency etc. It looks like if we can localise these desktops, we may be able to create environment for local non-IT users to interact with the computer for performing their tasks. The text used in menus and messages is in natural languages so it is assumed that user will not have any difficulty in understanding the environment once the menu as well as messages are in the language he is familiar with say Hindi, Tamil, Marathi.. It is thought that once these desktops are localised, the computers can reach more and more people, as people who do not understand English and therefore could not use computers are provided with an enviroment to work with computers.

According to me this is where we have failed to understand 'localising user environment'. We all know that we human beings communicate very effectively using natural languages. Language is not just an input/output mechanism; ability to read or write or speak and listen. It is a representation medium. Here, I would like to quote Putnam from Representation and Reality.

"In a very real sense, language is just an extension of our culture, a happy convention for the purpose of mutual communication. This shared context is essential for communication, it enriches it, and it limits it. Language is not merely an outgrowth of our culture, a convention for living in that culture. It is a fundamental determiner of our perspective. Our thinking is linguistic. Language is not only a medium for external communication, but for internal communication, planning, decision making and problem solving at conscious level."

Language builds our conceptual base through which we understand things. Words in a language are stable units standing for the concepts that make communication possible. If computers are to be used as agents for communication, we need to build this complex, heterogeneous medium for universal communication having shared context for both humans as well as computers.

Currently, the interfaces to the software developed on computer are very much implementation-dependent. Efforts are going on to build natural-language-like user interfaces on top of these systems to make them accessible to a wide variety of users. There are two problems:

Firstly, a natural language description used 'as it is' for interfaces that needs to be localised presents many problems. The problems can be characterised as problems due to ambiguity, problems due to fuzziness of symbols, problems with context sensitivity and problems with idiosyncrasies of the language (too many usages).

Secondly, a computer-based system after all, represents some real-world situation while the messages, interfaces are based on the implementation.

We propose following approach to localising user environment.

Our approach:

1. Building of representation scheme on computer plane:

   In my thesis [Irani, 1996] I have attempted characterisation of representation schemes. Representation schemes, internal language of mental representations, spoken language and written languages are built incremently, each one having its own plane, having what we call "concept base". Each succeeding plane provides a mapping for concepts from the earlier plane; in addition to that it has its own "vernacular" concepts. It follows from this that if computer systems are to be used for knowledge representation, the computer plane should also be incrementlly built.

The plane has to be built on the top of earlier planes.

2. Replacement of earlier descriptions based on ill-defined symbols on computer plane by something based on this representation scheme.

   The descriptions that form the user environment should be composed from the 'vocabulary' of this plane. (This essentially involves writing User Interface elements using what we call 'concepts' with compositional semantics so that communication is explicit, unambiguous and universal.)

3. Semi-automation of translation of localisation messages.

   Most of the messages are repetitive and based on a few most common concepts. A tool can be made available to have a crude translation of these messages from one language to another.

4. Building 'knowledge level' model for each software agent using the representation scheme.

   Knowledge is a description of the world. Representation is a way knowledge is encoded. From the 'Open source' we only come to know about the representation. While a user needs information at the 'knowledge level'.The term 'Knowledge level' was introduced by Newell [Newell,1982] to describe a system/agent as if it possesses certain knowledge.

   Without making commitments about representation or implementation issues more precisely, a Knowledge level model , according to Newell , is a model of behaviour in terms of the Knowledge and goals the agent has and the actions the agent can perform. The agent is driven by the principle of rationality, that is, it selects actions that it expects will lead to the satisfaction of its goals. Such a knowledge level model is essentially aimed at explaining why the agents behave in certain ways.

Our work and experience with localisation:

We have built a computer plane. Words in the vocabulary of a language (say English) are more or less expressions standing for individual concepts. We don't take just a 'word' as a concept, but a word in the language , in a particular context, in a particular role, in a particular domain, in a particular plane and having a primitive word associated with it as a 'concept'.

We have built a vocabulary of 5000 concepts for this purpose.

As we have already mentioned, using natural language for compositions in computer systems presents many problems. The problems can be characterized as problems due to ambiguity,

problems due to fuzziness of symbols, problems with context sensitivity and problems with idiosyncrasies of the language (too many usages).

In English, for example, there is no clear correspondence between words, their grammatical categories, their syntactic roles(place) in a phrase or in a sentence and their functions within a phrase or within a clause.

Words in general have many meanings. Some words can belong to different grammatical categories. In the absence of syntactic markers for the roles and separators and linkers for parts of speech, processing English mostly depends upon human beings ability to make 'sense' out of the construction.

The problem of getting roles of the constituent phrases is difficult, because in a sentence, the structure is flattened. There is a mixing of boundaries. A participant in a sentence can be a head word, which has a part to play as one of the roles in a sentence, or it is a modifier to one of the head words. Most often, it is possible to get the roles of the participants correctly if the modifying symbols belong only to one grammatical category(adjectives in case of nouns and adverbs in case of verbs) and therefore can be recognized syntactically.

In spite of these problems with natural languages, we feel that in general, a natural language is a relatively efficient and accurate encoding of the information it conveys. What makes it difficult to accept as a semantic theory is 'ambiguity'.

We postulate here that Streamlining and disciplining natural language can make it a good semantic language. The requirement of compositionality can be met if the syntax of a Natural language can be used for semantic compositions in the streamlined language [Irani, 1995].

Instead of devising an altogether new language, which people have to learn from scratch, we select an existing natural language to start with and streamline it to suit our purpose to keep the shared context intact.

We have a parser that works on Singlish and attempts to give a unique syntactic representation of a sentence. In case of failure it accepts input from the user.

We have analysed around 200,000 gnome desktop messages and menu items and separated the words in different categories:

1. Words to be translated in another language
2. Words to be transliterated
3. Words that are acronyms
4. Words that stand for short-forms
5. words that are on computer plane
6. Words that are culture dependent

Study of descriptions reveals that localisation becomes problematic as many of the messages(descriptions) are meant for the writer of the program who has in his mind the model of the program as well as its representation (encoding) on the computer.

Translations become difficult because the situational context is not known.

Identification of the part-of-speech is difficult; Words in general have too many meanings. Some words can belong to different categories. The syntactic roles taken by the elements can be found by the word order. However, these positions are not absolute.

Conclusion:

We strongly believe that the language for data descriptions as well as for messages should be based on the representation scheme having conceptual basis in the natural languages. 'Natural language as it is' is not suitable for user environment. Streamlining a natural language essentially adds an extra layer of annotations that help in disambiguating the flat structures of descriptions (sentences).

The need for localisation on a large scale and new challenges we face thereof can only be met by having user environment respecting 'semantics'. Canonical and unambiguous representation of messages and other user environment items is crucial for the world that hosts more than 6000 languages.

References:

http://www.ncst.ernet.in/projects/indix

http://www.ncst.ernet.in/projects/janabhaaratii

[irani, 1992] Interscript -A "script" for computation and communication of Indian languages across contexts, Alka Irani and Sylvia Candelana de Ram (Comp. research lab. New Maxico State Univ.), In Akshara- national seminar on information technology application in Indian Languages at Bhuvaneshvar, India, 1992