Integrated Project Management Software - Issues With Localisation (Part II)

By Volker Bendel

Article Word Count: 838   [View Summary] Comments (0)

Having a fully integrated job costing and accounting system, that works perfectly in one country, should make it relatively straightforward to employ it in other countries as well. After all most modern operating systems are able to provide a more or less seamless international and multilingual functionality on the touch of a few buttons.

In particular for a multinational company having easy access to all of their systems across borders is an important aspect of management. The mere translation of the project management interface is indeed quite an easy part of the localisation as most languages have words for "project" or "sales invoice" etc. There are however two main issues that will arise, one due to the integration of accounting and one due to the multilingual interoperability of the system: This second part of the article will look at the multilingual accessibility and interoperability issues after the accounting and bookkeeping issues have been looked at in Part I, published earlier on this forum:

These issues encountered when localising a software have to do with the users' flexibility to themselves change the interface. If the software has the ability for the user to rename fields in order to - for example - call a "project" a "job" or name a "client" a "customer", you would want this change to take effect throughout the company. Rather than changing the client interface on each single workstation, a commonly used option is to change those details in the data base to which every workstation logs on. The result of this setup is a mixture of where field names and program names come from. The "Select" in "Select Client" or "Select Job" would come from the local software whilst the words "Client" and "Job" would come from the data base to allow for companies to have a "Select Customer" or a "Select Project". The localisation for another language will therefore involve translation of the word "Select" as part of the software and the word "Client" as part of a standard database for that language (that may then be further amended by the end user).

Whereas this solution will work fine where the localised integrated project management system is only used in one and the same language, it will create additional snags if users of different languages need access to the same data. If for example group companies in different countries use the same job costing system and need to be able to access each others data or if the company's headquarter needs access to audit the data of the group companies you might need to log-on with an English client software into a Portuguese data base ending up with a mixture of languages in program- and field-names.

There are different approaches thinkable to resolve those issues:

First and Easiest - take away the flexibility of the users to make changes to the terminology of their system and base it all in the local installation

Pro:
-quick and easy to implement
-foolproof
-easy cross-language access

Contra:

-reduction of flexibility means reduction of user acceptance and conceived user-friendliness

Second - store all terminology in multiple languages both of data records as well as programs in the data base and only keep the executable on the local workstation with the executable giving a choice which language to use

Pro:
-relatively easy to implement
-easy cross-language access
-full flexibility is retained

Contra:
-increase of data traffic will most certainly affect the speed of all functions

Third - store all terminology both for data records as well as programs in the local installation with a choice to use the preset terminology, the user defined terminology or a different language

Contra:
-most long winded to implement

Pro:
-full flexibility retained
-speed not affected
-fully multilingual

This last option is most certainly the optimal choice when introducing multilingual functionality and working on the localisation of an integrated job costing system. If there is then an option to distribute terminology changes to all workstations via the data base, giving the user the option to update their software, when they log on to the data base or keep their default and in addition an option to revert to the preset settings, both flexibility and user acceptance as well as multilingual functionality are guaranteed.

Annex: This article has only covered some of the basic multilingual difficulties encountered when localising a project management software for other countries. It is therefore not meant to give a complete synopsis of every aspect of multilingual interoperability. Two of these aspects not considered, that would constitute even greater challenges to be overcome are 'Grammar' and 'Writing not based on the latin alphabet': If the software not only has program commands consisting of nouns, e.g. "Client Creation" or infinitive noun combinations, e.g. "Enter Job number", but more complex sentences, e.g. "Click here to open client entry screen", grammar in different languages would have to be considered. If the software needed to be localised for the e.g. Arab, Hebrew or Chinese language and writing, a whole lot of additional hurdles would need to be overcome.

© 2008 Volker Bendel - Volker Bendel is manager of the training department of Agency Software Worldwide, the producers of the "Paprika/Rebus" job costing software (http://www.paprika-software.com and http://www.rebus-software.com). Originally from a legal background, he has several years experience in planning and implementing Job Costing and Accounting Software Systems in the Creative Industry. He has also delivered training courses in the UK, Europe, Dubai, the US, China and Australia. Prior to that he worked as a senior business consultant in Hong Kong and as a department manager of a design department in Hong Kong.

Article Source: http://EzineArticles.com/?expert=Volker_Bendel

This article has been viewed 11 time(s).
Article Submitted On: September 13, 2008