1. Introduction

As can be seen in the many other papers in this book, decision making using DSS occurs in all countries and is undertaken by people who are not necessarily fluent in any language other than their mother tongue. Thus the decision support software must be made available in this mother tongue. But the situation is more complex than that, for some decisions may need to be taken using information gathered from across a complete region from data sources in different languages and cultures, and the decision making process may also involve many people from similarly diverse languages and cultures. The processes of collaborative decision making are covered in Chapter 12 on business negotiations, while in this chapter we will focus on how to make software available for a range of languages and cultures.

Globalization of commodity software products like word processors, spreadsheets, and their underlying operating systems, is becoming widespread. The suppliers of these systems, like Microsoft, Lotus, Claris, Apple, the main platform suppliers, and many others, have recognized since the start of the 1990s that more than half their revenues must come from outside the US and the English speaking markets. The US market is saturating.

This has meant that the manufacturers of software have developed methods for translating their products from their original target market, typically the US, to new markets. There are a wide range of issues to be addressed – these are described in Section 2.

The methods of attack have been based on relatively simple methods, on characterizing the essential features of a market within a “locale”, using a suitable character coding standard, and using resource files to factor out the locale dependent data like
The methods of attack have been based on relatively simple methods, on characterizing the essential features of a market within a “locale”, using a suitable character coding standard, and using resource files to factor out the locale dependent data like messages so that they can be easily replaced. These current approaches are described in Section 3.

Nevertheless, translation of software remains expensive, and not all markets are large enough to warrant translation. Thus, for example, the Same language of Lapland in Northern Finland, Sweden, and Norway, will not have the interfaces of office products translated, though the ability to store and manipulate data in the Same language will be enabled. Same is closely related to Finnish.

New approaches are needed, and I have been involved with some of these within the EU-funded Glossasoft project. These new approaches aim to exploit software architectures to factor out the locale dependent aspects behind Application Programmer Interfaces, and linguistics to enable this and avoid internationalization imposing unnatural interfaces upon the software. These approaches are described in Section 4.

Finally, in Section 5, we look forward to the kind of work that will be necessary to enable Decision Support Systems to succeed across countries and regions of the developing and developed world. All current approaches are aimed at single languages, though the same principles would work for a single system accessed through multiple languages, such as a decision support system spanning several countries or locales. There are two levels of linguistic issue: diverse interfaces in multiple languages accessing a single common repository of data which is in some manner language neutral; and systems where the shared data is in multiple languages intermingled. But behind all this there are deeper cultural issues about how decisions are made: social relations, and the perceptions of space and time, are critical. This area is only partially understood, and the issues and some indication of how to handle these technically will be discussed.

2. The localization problem

In moving any piece of software from one part of the world to another, it is important that the different needs of the users at these different locations are taken into account. These differences are:

1. the language of the user interface,
2. the language and number representations and measurement units of the data stored within the system,
3. the language used for product support
4. “cultural” factors like representation of currency, dates and colours
5. local practices such as legal requirements

These factors are also equally important when a single system is used through multiple languages, as when sharing information between different countries, Even