

Case Study

Flexpansion



Company overview

Background Flexpansion was founded on a predictive text expansion system which is based on computational linguistics PHD research at Edinburgh University by Dr Tim Willis. As well as predicting words from their initial few letters like conventional word prediction methods, this novel innovation will also take 'txt-spk' type abbreviated input and turn it into full text. Flexpansion's predictive text is also more advanced, having the ability to predict words from word frameworks as opposed to words typed in the correct letter order.

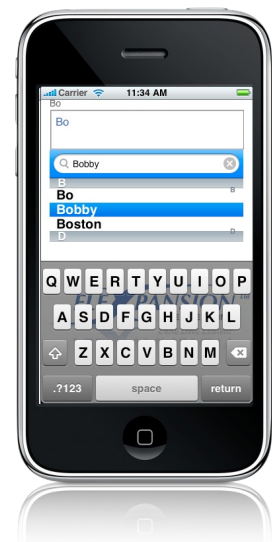
For example: 'g8' would equate to 'great,' or typing p,r,f, would suggest performance, perfume, etc. This gives scope to increase the speed of typing whilst producing the full formal English result which is required for business emails. Flexpansion could also be useful for people who struggle with spelling, order of nouns etc.

Web site: www.flexpansion.com

Feasibility Study

Flexpansion have the system working in a Java and C# version for use on PCs. The main aim of this feasibility study was to investigate whether the system could work on the iPhone by converting it to C, C++ or Objective C (the native iPhone language). The key areas of concern were the speed at which Flexpansion would run on a hand held device and whether the vocabulary data could be slimmed down to be workable with the devices limited memory. The deliverable was a proof of concept version of the Flexpansion software working on a stand alone application on the iPhone.

The feasibility study required a prototype to be produced. The development of the prototype was split into two parts - the GUI (graphical user interface) and the Flexpansion engine. Part of the investigation when developing the GUI involved looking at low level access to the keyboard functionality and accessibility to the devices inbuilt dictionaries. A number of different methods were discussed with regards to the engine including running it as a web service, cross compilation of the Java code, running a Java virtual machine on the device and a direct conversion of the Java code to Objective C.



The outcome

As mentioned the two key parts of the prototype development involved the GUI and the Flexpansion engine. Low level access to the keyboard functionality and dictionaries is currently denied but is under review by Apple. The GUI is fully functional and also has the addition of email for the user to email there message directly from inside Flexpansion.

After researching the problem it was decided that porting the code to objective C was the most feasible solution, albeit a time consuming one. The code conversion was completed, but as this used up most of the feasibility study time, Flexpansion still require the code to be tested, work-a-rounds to be put in place for unusual Java methods and testing for efficiency and memory leaks to be carried out. One key progression of the study was to highlight the use of a database as the vehicle for holding the word data. The data is currently stored in a large xml file. Loading the file would have required some clever memory mapping to ensure a big enough area of contiguous memory was available to load all the data. Hence the database would appear to be a better long term solution.