



QUESTION

"Is your multimodal software better than mine?"

How to compare multimodal modules from a practical software engineering point of view

AUTHOR

Jean-Claude MARTIN
LIMSI-CNRS, France, martin@limsi.fr, www.limsi.fr/Individu/martin

BACKGROUND

Several multimodal modules combining several input modalities such as speech and gesture have been already implemented (cf CMC'98, IJCAI'97, CMC'95, IMMI95). Comparison of these multimodal modules can be done on the basis of their evaluation during user studies. Yet, user studies are done with different protocols from one multimodal module to the other. Hence comparison is difficult. Multimodal systems can also be compared on the basis of the number and complexity of modality they process. Yet, it seems weird to take into account the monomodal aspects of multimodal modules in their comparison.

In this paper, we suggest two new criteria for comparing multimodal modules from a software-engineering point of view: how well do these multimodal modules address the upgrading and composing problems.

The upgrading problem. Most of today's applications combine a graphical screen, a keyboard and a mouse. With the success of mobile heterogeneous configurations and the availability of several speech recognition developer's kits, solutions have to be found on how to access such existing mono-modal applications through other input media and other modalities such as gesture and speech. Thus the question to ask to a multimodal developer concerning the methodological upgrading problem is "how would you integrate your multimodal module into an existing monomodal application in order to upgrade this application to a multimodal application?"

The composing problem. Component-based environments for developing graphical user interfaces are becoming more and more widespread (i.e. JavaBeans). With these environments, developers can combine existing graphical components specified by their properties (i.e. size, color for graphical components such as buttons) and the behavior of such components (i.e. reaction to events and generation of events). Existing prototypes of multimodal interfaces are monolithic in the sense that they do not include such model of multimodal component. Thus, the question related to the composing problem is "how a developer could integrate your multimodal interface with other interface components (including graphical user interface components, dialog components, mono or multimodal components)?"

PRELIMINARY IDEAS

We will present some preliminary ideas on how to address these questions and illustrate them with the model and system of multimodal interface that we have developed. The methodology for upgrading a monomodal application to a multimodal application with our model is the following: 1) extract the application model out of the existing mono-modal application, 2) specify the set of referencable objects of this application (i.e. objects that the user will be able to refer to with speech and/or gesture, 3) specify the speech and gesture monomodal processing and 4) specify the types of cooperation between modalities and link them to the application model. The composing problem is also addressed thanks to the modularity of both the specifications of referencable objects and the specifications of the cooperations between modalities. Yet, problems remain regarding the conflicts that may occur between two multimodal components, and also problems related to multimodal commands involving several multimodal components.

POTENTIAL DISCUSSION

Discussion may help in finding out whether the *upgrading* and *composing* questions are indeed interesting for the comparison of multimodal modules from a software-engineering point of view. It may also be useful to know how multimodal developers present at the workshop address these two questions in their own work. We believe that such questions may also help to focus on some core features of multi-modality.