Halbrügge, M. (2013). ACT-CV: Bridging the Gap between Cognitive Models and the Outer World. In Brandenburg, E., Doria, L., Gross, A., Güntzler, T., and Smieszek, H., eds., Grundlagen und Anwendungen der Mensch-Technik-Interaktion. 10. Berliner Werkstatt Mensch-Maschine-Systeme, pages 205-210, Berlin, Universitätsverlag der TU Berlin

ACT-CV: Bridging the Gap between Cognitive Models and the Outer World

Marc Halbrügge[1]

Keywords: Cognitive Modeling, Human Computer Interaction, Usability, Prototyping

Abstract

Cognitive modeling can provide important insights about the usability of software or hardware products at all stages of product design. It is not used very often, though, as the creation of cognitive user models is typically done by highly trained experts. These specialists need not only develop the models, but have to spend a lot of additional time on software interface problems: The cognitive models need to be connected to the interactive system under evaluation. This paper presents ACT-CV, a software library that aims at solving the interface problems of user modeling especially in usability research.

Introduction

Cognitive modeling has proven more and more useful for the design and evaluation of human machine interfaces as it allows to predict usability metrics like the time to complete a task or the types of errors that may occur. The psychological basis for cognitive user models is laid by cognitive architectures like ACT-R (Anderson, 2007). ACT-R is a software framework that provides a well-researched environment that closely mimics the cognitive, perceptual, and motor capabilities of human users. Unfortunately, cognitive architectures tend to lock the modeler into highly specialized computer systems (Lisp in the case of ACT-R), creating significant hurdles for the application of a user model to the human machine interface in question (Byrne, 2013).

In the case of basic psychological research, like visual search experiments, the user interface (UI) can be created within the system, e.g. implemented using Lisp (see also Urbas et al., 2006). This is not an option for existing software that already has been developed, and where Lisp is highly uncommon or in cases where changes to the original software are not possible.

CogTool (John et al., 2004) is an extremely powerful tool for creating user models, but it only works on mock-up prototypes of the user interface under evaluation, i.e. it only eases the lock-in problem, but doesn't solve it. A different approach is taken by SegMan (Ritter et al., 2006) which identifies elements on the screen based on pre-defined pixel color comparisons. SegMan's major advantage is that it al-

[1] Quality and Usability Lab, Telekom Innovation Laboratories, Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, marc.halbruegge@telekom.de

lows to break out of the Lisp system, but its screen processing capability is rather hard to use and cannot handle dynamic screens very well.

The software library ACT-CV (Halbrügge et al., 2007) uses a similar approach as SegMan for allowing direct interaction between ACT-R and arbitrary software. It differs a lot on the technical side, as it uses computer vision for the identification of screen elements and adds support for time based visual parameters like optical flow. The graphic processing inside ACT-CV is handled by the open source framework OpenCV (http://opencv.org/) which provides state of the art machine vision on multiple platforms and includes more than 2500 individual algorithms.

The computer vision approach allows the extraction of much more complex features of the computer screen than just the color of a given pixel. In addition, it provides access to the external world using a video camera.

ACT-CV was used for modeling visual tasks especially in the automotive domain, where it allowed to connect a cognitive model of car driving (Salvucci, 2006) to an already existing driving simulation (Deml and Neumann, 2008; Halbrügge et al., 2008, see figure 1). The applications of ACT-CV, while successful, have also brought some weaknesses and other issues into light. This paper is about how these issues are addressed by an upcoming new revision of the software library.

Figure 1: Driving simulation experiment from Halbrügge et al. (2008)

Application and Shortcomings of ACT-CV 1

The simple task of finding the position of a predefined visual element (e.g. a search button) on the computer screen can be accomplished by using ACT-CV out of the box and providing nothing but a picture file containing that visual element to it.

But when the task becomes more complex, the usability of ACT-CV degrades sharply. A potential user of the system has to extend the software with his or her own machine vision code in order to apply it to new interfaces. While recompiling the system needs only average software development skills, designing the sensor code that provides information from the outer world to the cognitive model demands some knowledge of computer vision algorithms as well.

Another problem of ACT-CV 1 is the lack of text processing. Most interfaces rely at least partly on written language. ACT-CV in its first incarnation comes with many sensors for colors and shapes, but it cannot decipher text. This shortcoming is hard to overcome, e.g. the sensor for the visual search on road signs task in Halbrügge et al. (2008) had to be implemented as a comparison of pictures of letters, which is an extremely CPU intensive and error prone workaround. Figure 2 demonstrates the signal flow that was used in this experiment.

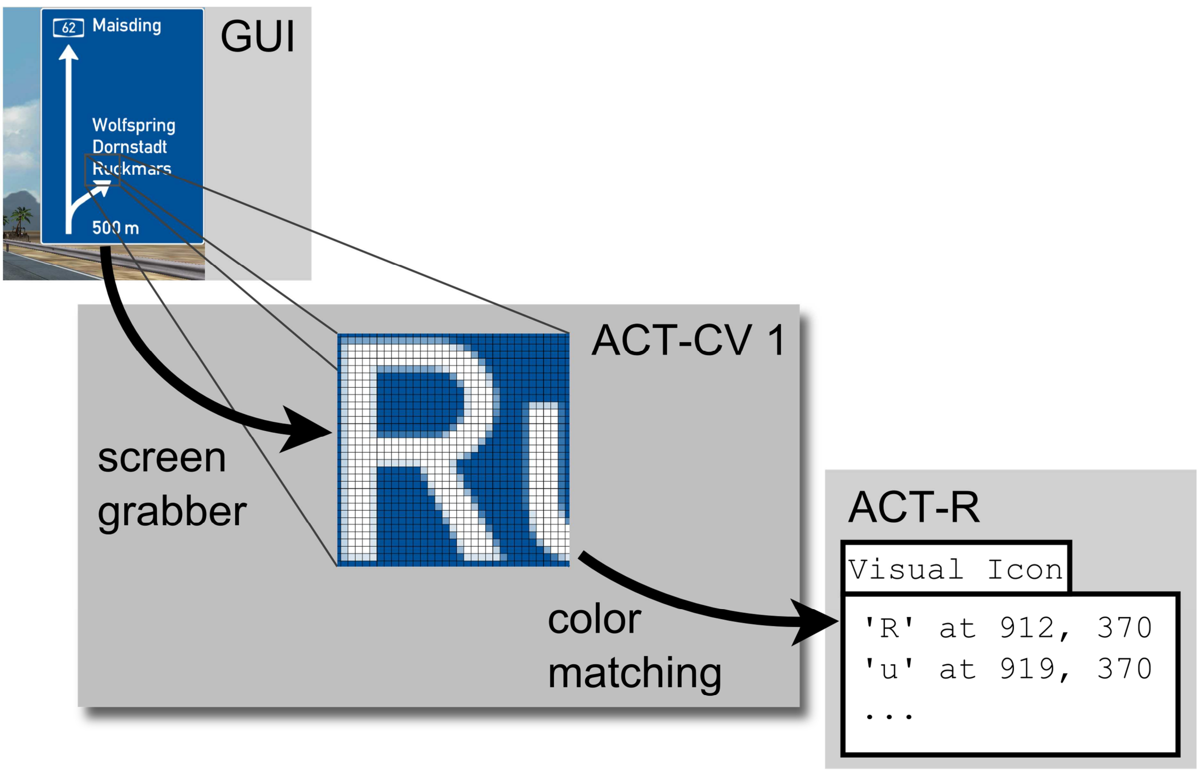

Figure 2: Application of ACT-CV 1 to the road sign reading task

ACT-CV 2

ACT-CV 2 is a major overhaul of the system. It simplifies the use of the library and at the same time addresses the lack of text processing in the previous revision of ACT-CV by adding HTML5 support. HTML5 (Hickson and Hyatt, 2011) marks the progression of HTML from a markup language for static text documents to a programming language for interactive user interfaces. It powers not only the web, but many native mobile applications, and is used as platform for rapid prototyping as well. ACT-CV 2 features a state of the art web browser that allows both human subjects and cognitive models to interact with any HTML5 application.

It uses the open source WebKit (http://www.webkit.org/) rendering engine for both creating the GUI that is presented to the user and for extracting the (textual) features that are handed over to ACT-R (see Figure 3). The pixel based computer vision approach of ACT-CV 1 is still available, allowing the combination of symbolic text features extracted from the HTML document (DOM) with sub symbolic visual features like bottom up salience or optical flow extracted from the rendered user interface.

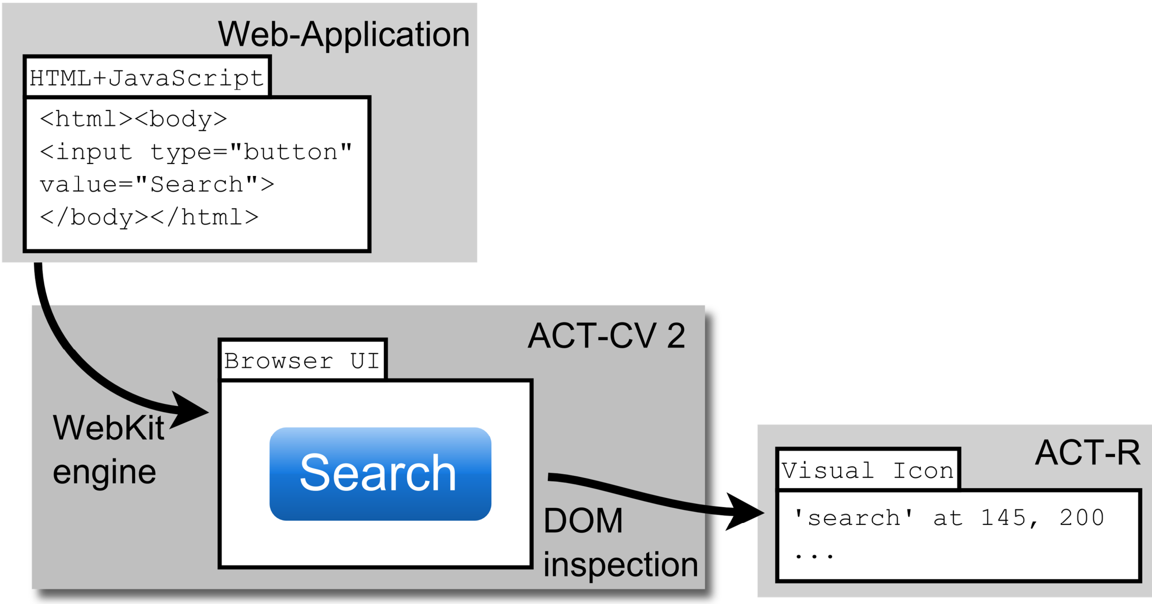

Figure 3: Signal flow for HTML5 applications in ACT-CV 2

Planned Applications of ACT-CV 2

The first application of ACT-CV 2 will be the usability evaluation of an HTML based cooking assistant that has been developed as part of the model based user interface project MASP (Blumendorf et al., 2008). ACT-CV will be used in conjunction and comparison with the MeMo workbench for automated usability evaluation (Engelbrecht et al., 2009; Quade et al., 2013).

MASP automatically adapts the UI of the cooking assistant to the different devices that it supports (e.g. smartphone, TV). This creates the need to be able to test the usability of the UI for each variation in layout and size. Covering all those variations using CogTool or SegMan would create significant overhead as one had to address all of them individually. ACT-CV 2 on the other hand can handle all variations at once as long as the wording on the screens doesn’t change.

Future Extensions

Future extensions of ACT-CV will probably include automatically generated saliency maps based on the graphical input to the system. This will form the basis for more accurate models of visual search, as needed for example to predict the probability to overlook a button in a given UI (Ruß, 2011). The hybrid approach of

ACT-CV 2 is well suited for this extension as OpenCV already comes with bio-inspired algorithms for foveal vision (Benoit et al., 2010).

Another interesting extension is the integration of Java ACT-R (Salvucci, 2010). As ACT-CV uses Java for accessing the screen and translating mouse and keyboard events generated by the cognitive architecture to actual events of the operating system, the inclusion of Java ACT-R should not create too much effort. It would substantially ease the creation and maintenance of cognitive user models.

Conclusions

In order to use cognitive modeling during the usability engineering process, one has to combine the model on the one hand with the interactive environment of the software in question on the other hand. Creating interfaces between both is technically challenging. ACT-CV, the software library presented here, aims at bridging the gap between cognitive models and the outer world. It provides access to any graphical user interface by the means of machine vision applied to the current content on the computer screen. But the universality of ACT-CV's approach comes with the downside of demanding basic software engineering and computer vision knowledge.

ACT-CV 2, the upcoming revision of the library, adds new features while easing its use by adding native support for HTML5 applications. In the special case of HTML based user interfaces, ACT-CV 2 allows to bypass the computer vision layer and provides the textual elements of the UI directly to the vision module of the cognitive architecture. This new approach renders laborious tasks like manually annotating screenshots of the UI unnecessary and allows the modeler to fully concentrate on the cognitive user model itself.

ACT-CV is available for download at http://act-cv.sourceforge.net

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press, New York.

Benoit, A., Caplier, A., Durette, B., and Hérault, J. (2010). Using human visual system modeling for bio-inspired low level image processing. *Computer vision and Image understanding*, 114(7):758–773.

Blumendorf, M., Feuerstack, S., and Albayrak, S. (2008). Multimodal user interfaces for smart environments: the multi-access service platform. In *Proceedings of the working conference on Advanced visual interfaces*, pages 478–479. ACM.

Byrne, M. D. (2013). Computational cognitive modeling of interactive performance. In Lee, J. D. and Kirlik, A., editors, *The Oxford Handbook of Cognitive Engineering*, chapter 27. Oxford University Press, New York.

Deml, B. and Neumann, H. (2008). Predicting the distraction of car drivers by cognitive modelling. *ATZelektronik*, 3(2):14–17.

Engelbrecht, K.-P., Quade, M., and Möller, S. (2009). Analysis of a new simulation approach to dialog system evaluation. *Speech Communication*, 51(12):1234–1252.

Halbrügge, M., Deml, B., Färber, B. A., and Bardins, S. (2007). ACT-CV – Die Erweiterung von ACT-R um Bildverarbeitungsalgorithmen erlaubt die schnelle Erzeugung mächtiger Benutzermodelle. In Grandt, M. and Bauch, A., editors, *Simulationsgestützte Systemgestaltung* – DGLR-Bericht 2007-04, pages 313–331, Bonn. DGLR e.V.

Halbrügge, M., Deml, B., and Neumann, H. (2008). A cognitive model of visual search on direction signs alongside the autobahn. *International Journal of Psychology*, 43(3/4):759.

Hickson, I. and Hyatt, D. (2011). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. W3C Working Draft edition.

John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K. (2004). Predictive human performance modeling made easy. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 455–462. ACM.

Quade, M., Lehmann, G., Engelbrecht, K.-P., Roscher, D., and Albayrak, S. (2013). Automated usability evaluation of model-based adaptive user interfaces for users with special and specific needs by simulating user interaction. In Martin, E., Haya, P. A., and Carro, R. M., editors, *User Modeling and Adaptation for Daily Routines: Providing Assistance to People with Special Needs*, pages 219–247. Springer, Berlin / Heidelberg.

Ritter, F. E., Van Rooy, D., Amant, R. S., and Simpson, K. (2006). Providing user models direct access to interfaces: an exploratory study of a simple interface with implications for hri and hci. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(3):592–601.

Ruß, A. (2011). Modeling Visual Attention for Rule-based Usability Simulations of Elderly Citizen. In Harris, D., editor, *Engineering Psychology and Cognitive Ergonomics*, pages 71-81. Springer, Berlin.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. Human Factors: *The Journal of the Human Factors and Ergonomics Society*, 48(2):362–380.

Salvucci, D. D. (2010). *Cognitive supermodels*. Invited talk, European ACT-R Workshop, Groningen, The Netherlands.

Urbas, L., Heinath, M., Trösterer, S., Pape, N., Dzaack, J., Kiefer, J., and Leuchter, S. (2006). Agimap: A tool-chain to support the modeling of the interaction level of dynamic systems. In *Proceedings of the 7th International Conference on Cognitive Modeling* (S. 409). Trieste, Italien: Erlbaum.