End-User Programming of Ubicomp in the Home

Nicolai Marquardt
University of Calgary, Department of Computer Science
Course 701.81 – Domestic Computing, Prof. Saul Greenberg

INTRODUCTION AND MOTIVATION

With the increasing number of electronic devices and computers in homes to support everyday domestic activities, it is very complicated for users to adapt and modify this technology to their personal requirements. Therefore, researchers have applied the research of end-user programming (EUP) to ubiquitous computing (Ubicomp) in the home.

END-USER PROGRAMMING STRATEGIES

End-user programming systems try to make the abstract and high-level concepts that are required to program computers understandable for non-expert users. These systems lower the entry barrier for users (*low threshold*), but also try to provide powerful and flexible functionality (*high ceiling*) to create new envisioned applications [5]. Many of these systems implement one of the following strategies for end-user programming:

- *Simplified programming languages*:
  This approach is making programming languages easier to understand by using words of the everyday language instead of specific keywords. However, the level of abstraction is still very high, and users need to understand high-level programming concepts.

- *Visual programming systems:*
  These systems provide graphical visualizations of the program functionality [4]. Usually these are visualizations of fundamental building blocks that can be connected by the end-user to construct the desired program behavior. These visualizations, however, can become very complex and confusing for more sophisticated programs.

- *Natural language interpretation:*
  With this approach, the end-user programming systems try to infer the desired program directly from the users' instructions. These systems are often implemented as a dialog between the computer and the user, and sometimes combined with the *programming by demonstration* technique. However, it is very complex for users to modify existing programs afterwards; therefore, the systems are sometimes provide a graphical feedback similar to the *visual programming systems*.

- *Programming by demonstration/example (PBD/PBE):*
  With this approach, users can directly demonstrate parts of the program behavior (e.g., triggering conditions, requested response) and the end-user programming system build the necessary application. Similar to the *natural language interpretation*, feedback of the inferred application is very important (and can be implemented as live previews of the program behavior, textual output of the program functionality or graphical visualizations).

END-USER PROGRAMMING IN THE HOME

The following EUP systems allow non-expert people in the home to change the pre-defined program behavior of Ubicomp applications. Truong et al. [6] implemented a system that lets users specify the desired program behavior by building sentences similar to *magnetic poetry*. The system infers the application similar to the mentioned *natural language interpretation*. In their studies they also observed that people usually do not think about specific hardware devices when creating new applications with the system. The Media Cubes allow the programming with *tangible* objects [1]. The cubes can be combined to build a new application. The *a CAPpella* system [2] lets users create new context-aware applications *in situ* with the *programming by demonstration* approach. Humble et al. [3] describe an end-user programming system with a graphical user interface. The system uses the metaphor of jigsaw pieces that the users can combine to create new ubiquitous computing applications.

These systems illustrate various methods of how people with no programming knowledge can create Ubicomp applications in the home (*low threshold*). The challenge, however, remains of finding EUP systems with the same low entry barrier, but higher flexibility of the created applications (*high ceiling*).

IMPORTANT REFERENCES

1. Blackwell, A. F. and Hague, R. (2001) AutoHAN: An Architecture for Programming the Home. In Proceedings of the IEEE 2001 *Symposia on Human Centric Computing Languages and Environments (HCC'01)*. IEEE Computer Society, Washington.
2. Dey, A. K., Hamid, R., Beckmann, C., Li, I., and Hsu, D. (2004) a CAPpella: Programming by Demonstration of Context-Aware Applications. In *Proceedings of the SIGCHI 2004*, pp. 33-40. ACM, New York.
3. Humble, J., Crabtree, A., Hemmings, T., Akesson, K., Koleva, B., Rodden, T., and Hansson, P. (2003) "Playing with the Bits" - User-configuration of Ubiquitous Domestic Environments. In *Proceedings of Ubicomp 2003*, pp. 256-263. Springer, Berlin/Heidelberg.
4. Myers, B. A. (1986) Visual programming, programming by example, and program visualization: a taxonomy. In Proc. of the SIGCHI 1986. ACM, New York, pp. 59-66.
5. Myers, B., Hudson, S. E., and Pausch, R. (2000) Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1, pp. 3-28.
6. Truong, K. N., Huang, E. M., and Abowd, G. D. (2004) CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home. In *Proceedings of Ubicomp 2004*, pp. 143-160. Springer, Berlin/Heidelberg.