Using Speech for End User Programming of Smart Environments in the Internet of Things

Thomas Kubitza
Human Computer Interaction Lab
University of Stuttgart
Pfaffenwaldring 5a
70569 Stuttgart, Germany
thomas.kubitza@vis.uni-stuttgart.de

Abstract
More and more of our daily environments already embed smart networked devices. However, most of these devices still work independently without collaborating with other devices that are for example in proximity. In *smart environments* networked sensor rich devices can be interwoven in order to optimally serve their users. Current End User Programming approaches however are hardly suitable for mobile contexts but instead bind users to classical desktop computers. The author argues that natural speech processing combined with the concept of programming by demonstration can be a promising way for a more mobile, intuitive and efficient orchestration of end user defined smart environment behavior. A mobile application named *Speechweaver* is presented that implements this concept on top of an existing smart environments middleware.

Author Keywords
Smart Environments, End User Programming, Programming by Speech, Mashups, Internet of Things

Introduction
We see a rapid increase of smart devices in our daily environments. Today most of these devices still process

their tasks individually without cooperating with other devices that are for instance in their proximity. Ubiquitous Computing research ever since investigates how smart interconnected devices equipped with sensors and actuators can cooperate in order to optimally fulfill more complex tasks for their users. More recent examples of this are smart home environments in which sensors and context data can be dynamically used by the system logic to control actuators (for example dimming the room light based on indoor and outdoor light sensors, the daytime and the presence and preferences of inhabitants). One popular way to realize such scenarios is allowing inhabitants to create their own user generated logic (End User Programming). So far the predominant way of realizing this concept are graphical user interfaces on classic computers or on mobile devices such as tablets and smartphones which allow users to define relations between sensors events and actuator actions in a graphical or forms based way. A popular example for latter is the if-this-then-that (www.ifttt.com) web service. However, through the wide availability and high accuracy of recent natural language processing software, services and hardware (e.g. Amazon Echo[1]) a novel interaction space is opened up that has the potential to make End User Development approaches for smart environments more intuitive and efficient.

The research of the author particularly investigates the use of natural language for programming the behavior of smart environments. Based on the authors earlier research in the creation of smart environments [4] a multimodal user interface for mobile devices named "Speechweaver" has been developed. Speechweaver runs on smartphones and tablets and takes spoken input such as *"Please always turn the light on when I enter this room",* instantly interprets it, turns it into an executable script and executes it in the smart environments middleware (presented in [4]). Simultaneously it provides visual feedback on the mobile devices screen in the form of tokens that represent the spoken text (Figure 1 left). This allows quick adaption of the spoken input in case multiple options are possible – for instance when in the context of "*this room*" multiple smart lights are available. In this case the GUI would offer a dropdown list of the available lights in the current context when touching the token "*the light*". The most innovative feature of Speechweaver is however the combination of natural language speech input with the concept of Programming by Demonstration (PBD).

Combining Speech Input with Programming by Demonstration

With the rapidly growing number of smart networked devices even in a single room it can become difficult to address devices that a user wants to use. For example if a user says *"Show me bbc.com on the display."* and multiple devices with a display are present the ambiguity needs to be resolved by the user. With the approach described above a second step would be necessary for instance by selecting one of the available displays using the dropdown list in the visual interface. However, our approach offers another option that allows addressing a desired device **while** speaking. Now a user can say "*Show me bbc.com on this display."* and while saying this shake his tablet or smartphone or actuates another sensor associated with the desired display. Speechweaver will then in an

[1] http://www.amazon.com/Amazon-SK705DI-Echo/dp/B00X4WHP5E

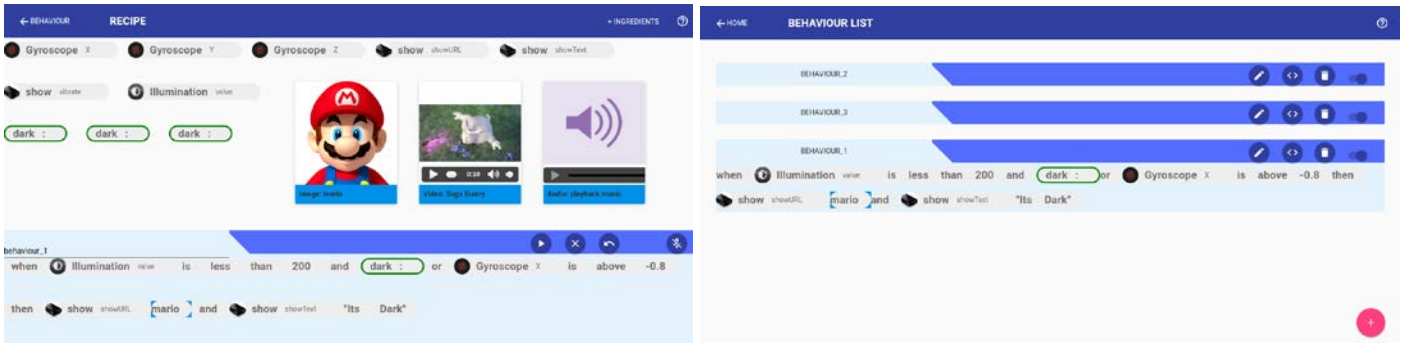

Figure 1:Speechweaver UI: Recipe editor with ingredients area in the top and speech tokens in the bottom (left), list view of existing recipes (right)

intelligent way correlate the manipulated sensor (in this case the accelerometer of the desired tablet) with the sentence spoken by the user. The result is that the webpage "http://bbc.com" is displayed on the screen of the tablet device that was shaken. This approach can be basically used in combination with any sensor that can be directly manipulated by the user. For instance a user could associate a wall switch with his desk lamp by saying "*Connect this switch with the desk lamp*" and while saying this the user would actuate the wall switch. As result the user would be able to permanently control his desk lamp with the wall switch until he decides to remove this user defined behavior. The creation and management of such user defined behavior is described in the next section.

Speechweaver Concept

Speechweaver is an application for mobile devices that has been developed to work hand in hand with the meSchup middleware [4] for smart environments. meSchup provides a common layer and unified access to all sensor and actuator equipped devices within a smart environment. Behavior that spans across many devices can be simply implemented in the form of Javascript scripts. This is where Speechweaver builds on top: Spoken input is instantly processed into a textual representation e.g. using the Google Speech API, translated into Javascript code and deployed in the smart environments middleware. Speechweaver is a mobile multimodal interface for end users that allows to program the behavior of smart environments in the context of the environment by using natural speech and optionally direct physical interaction with the environment.

The creation of a new behavior within Speechweaver is based on a simply analogy: *cooking*. When a certain recipe needs to be cooked the first step of the cook is usually the collection of ingredients and then in the second step all ingredients are combined to the final dish. Similarly Speechweaver allows a user to create a new recipe by firstly collecting the necessary components (e.g. sensor and actuator references, images, videos, URLS, etc.) and secondly formulating the logical relationship between the ingredients by using natural language. Using the PBD approach both steps can often (but not always) be combined in a single step (as described in the last paragraph). In the graphical interface the collection of ingredients is visualized in the upper part (Figure 1 left). In the lower part of the figure the latest speech input is shown using graphical tokens with the words of the sentence inside. References to ingredients that have automatically been matched are visualized within the tokens. Internally

ontologies are used to match device/sensor/actuator names and their relations with natural language. Each of the visual tokens can be deleted or new tokens can be dragged and dropped from the ingredients area into the tokens section. The intention of this is to allow quick correction if certain parts of the spoken input are not or wrongly recognized. The “play” button instantly indicates weather the current sentence can be transformed into a valid behavior script for the meSchup middleware. Valid behavior is instantly deployed and can be directly tried out by users in the smart environment. Multiple user defined recipes can be run in parallel (Figure 1 right), can be disabled, edited or deleted persistently.

The Speechweaver project has not been published yet and is still in active development. However, all functionalities described in this paper are already functional and can be demonstrated during the workshop.

Fit and Contribution

The author believes that the presented research perfectly fits into the theme of the workshop. It lies at the intersection of HCI, Ubiquitous Computing, Internet of Things and Natural Language Processing and touches very recent problems that are interesting for both academic research and commercial exploitation. Coming from the HCI side the author especially expects a rich exchange with the speech community in order to improve and extend his expertise in this field. At the same time the research of the author can inspire both speech and HCI researchers through its very practical and end user application oriented nature.

About the Author

Thomas Kubitza is Ph.D. candidate at the HCI lab of the University of Stuttgart lead by Albrecht Schmidt. His research is settled in the area of ubiquitous computing, specifically in distributed physical computing, smart environments and the Internet of Things. Thomas was organizer of multiple workshops targeting physical computing at conferences such as TEI [1], mobileHCI or IS-EUD [5]. Among others he has published in the area of Ubiquitous Computing [3][2] and End User Development [4].

References

1. T Kubitza, A Schmidt, and N Pohl. 2014. Tools and methods for creating interactive artifacts. *Proceedings of TEI ’14*. Retrieved January 26, 2015 from http://dl.acm.org/citation.cfm?id=2568483

2. Thomas Kubitza, Norman Pohl, Tilman Dingler, and Albrecht Schmidt. 2013. WebClip. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp ’13*, ACM Press, 387. http://doi.org/10.1145/2493432.2493520

3. Thomas Kubitza, Norman Pohl, Tilman Dingler, Stefan Schneegass, Christian Weichel, and Albrecht Schmidt. 2013. Ingredients for a New Wave of Ubicomp Products. *IEEE Pervasive Computing* 12, 3, 5–8. http://doi.org/10.1109/MPRV.2013.51

4. Thomas Kubitza and Albrecht Schmidt. 2015. Towards a Toolkit for the Rapid Creation of Smart Environments. In *5th International Symposium on End-User Development (IS-EUD ’15)*. Springer, 230–235. http://doi.org/10.1007/978-3-319-

18425-8_21

5. Thomas Kubitza. 2015. IS-EUD 2015 Studio : Exploring End User Programming of Interactive Spaces. *5th International Symposium on End-User Development (IS-EUD '15)*, Springer, 283–286. http://doi.org/10.1007/978-3-319-18425-8_30