Alan F. Blackwell, Saeed Aghaee (2015):
IoT Programming needs Deixis.
In International Reports on Socio-Informatics (IRSI),
Proceedings of the CHI 2015 - Workshop on End User Development in the Internet of Things Era
(Vol. 12, Iss. 2, pp. 49-54)

IoT Programming needs Deixis

Alan F. Blackwell[1], Saeed Aghaee[2]

University of Cambridge, UK
[1]*afb21@cam.ac.uk*
[2]*aghaeester@gmail.com*

Abstract. In this paper, we argue that the linguistic concept of deixis is an essential consideration in the design of end-user programming (EUP) languages for Internet of Things (IoT) applications. Deixis offers a theoretical account of how physical contexts can be incorporated in the semantics of information systems. By analysing several examples of EUP4IoT, we offer design guidelines for future systems in this area.

1 Introduction

Deixis is a linguistic term, describing situations in which the meaning of a sentence can only be resolved by reference to the physical context in which it is spoken. A typical example might be “pick that up now!” (said while pointing at an object on the floor). The meaning of the sentence is unclear without the parenthetical explanation that follows it in this text, and the precise interpretation is only available to a person present in the place where it was spoken, able to see the speaker's finger and the object at which it was pointing. This simple definition already introduces key concepts ofinteraction in ubiquitous computing. “Context” is both a key technical challenge in Ubicomp (e.g. inferring context from sensor data in smart environments), and also a key question in the critical philosophy of interaction for Ubicomp [5]. Deixis is similarly a challenging question for computer processing of natural language, because it draws attention to situations in which text alone is insufficient for semantic interpretation.

A further challenge is that computational models of mind do not always account for embodied action. It is necessary to appeal to theories of external representation [7], to describe the ways in which the pointing finger supplements working memory through persistent availability in a perceived scene.

2 Deixis in Tangible Interaction

These issues of context and representation have been addressed in the field of Tangible and Embodied Interaction (TEI). However, deixis as a feature of language in itself is relatively under-explored. In part, this is because the traditional command line and GUI is kept distinct from new tangible devices. Although designers of tangible systems use conventional UIs to implement them, studies of end users focus on the novel interaction modes.

When considering end-user programming, we must pay attention to deixis. Whereas tangible interaction is an especially direct type of direct manipulation, programming is fundamentally linguistic, rather than physical. When we write programs, we are making statements about the things we want to happen in the future. A key principle of direct manipulation is that the effects of one's actions should be immediately visible. In programming, by contrast, the effects of one's actions will occur in the future, once the program is executed.

Figure 1: Use of the deictic references in the tangible IoT programming language *Media Cubes*, designed for end-user scripting in AutoHAN [4].

One approach to programming in the context of tangible interfaces has been the creation of tangible programming languages [6]. The design intuition is that, if the GUI is more concrete and intuitive than the mathematical abstractions of programming, so 3D tangible UIs (TUIs) will be even more concrete and intuitive. Although it is not immediately apparent how physical objects can be a “language”, there is an appealing intuition that programming by assembling software components is like assembling Lego bricks. Many tangible programming languages have been created on the basis of this intuition.

3 Tangible programming languages

One example of a TUI that is particularly relevant to end-user programming of IoT is the Media Cubes language for scripting interaction between networked devices in a “smart home” automation scenario [3]. The Media Cubes language was motivated by analysis of domestic remote controls, refactoring them so that each control had only a single button. Some Cubes represented state change events (e.g. play/pause), while others represented media content or indexes (e.g. a video stream). Users could specify relations between the Cubes and a range of media devices by placing them in proximity while pressing a button. Scripts were built up by composing these associations. However, using physical objects both for deixis and for language syntax introduced problems In tangible programming languages such as Media Cubes, individual cubes function as a reference (after being associated with an appliance) and also as a relation (such as a media stream, index or event). Tangible representation of relations was valuable, because it allowed relations to be composed (for example, specifying that the occurrence of an event should play or pause a media stream). However, this emphasis on tangible representation of abstract entities removed the directness of interaction with the appliances themselves. A further problem of Media Cubes, often commented on at the time, was that tangible programming languages represented an extreme position in Cognitive Dimensions. Although placing cubes together and pressing a button could naturally compose scripts, the permanent record of that action was not visible, because the configuration of cubes would change as soon as the user put them away. When regarded as a language notation, physical objects have many drawbacks, as analysed in detail by Edge [6], but also anticipated much earlier by the satirist Jonathan Swift in Gulliver's Travels, where the philosophers of Lagado carry baskets of physical objects in order to speak in a universal language without any ambiguity of interpretation.

4 Integrating direct manipulation with deictic language

As an alternative to the creation of unimodal programming languages, whether tangible, graphical or textual, multimodal languages exploit the relative advantages of different modes in a way that is analogous to deixis in natural language. Just as the introductory example of deixis involved both verbal (speech) and physical (finger) components, so multimodal languages can use a mouse pointer alongside textual or spoken interaction. The classic example of deixis in multimodal interaction is the Put That There system [4]. Aghaee's Natural Mash [1] provides a powerful and intuitive paradigm for web mashups, allowing the user directly to demonstrate actions within a browser, and then compose these actions into a script through use of a constrained natural language in a textual dialogue window. In our current research, we are extending the interaction paradigm of Natural Mash to IoT, by incorporating real world interactions with internet-enabled devices into the repertoire of services that can be mashed up. However, whereas Natural Mash employs a form of deixis through actions in a (GUI) browser window, interaction with an Internet appliance takes place in the 3D physical world. It is possible to describe these actions indirectly, for example using the onscreen controls of a remote camera app to steer and zoom the camera. This extension is relatively straightforward, and offers an immediately accessible scripting extension to many typical IoT devices. But the argument presented in this paper suggests a more ambitious approach to the use of deixis, integrating the tangible affordances of the physical device with the potential for scripting.

Figure 2: Use of deicitic references to capture actions in NaturalMash [1].

5 Augmented reality as deixis

We have created a simple mobile augmented reality app that can be used to identify, refer to, and manipulate abstract functions that are associated with

physical appliances. In a sense, mobile augmented reality always represents a form of deixis - the phone camera becomes a pointing device, specifying a context within which some type of linguistic query or command might be interpreted [8]. Most such systems represent enhanced forms of direct manipulation, but we are interested in turning them into a form of programming language. A programming language can be considered as an abstract layer superimposed on the direct manipulation of data. In the same way, augmented reality can be considered as an imaginary world of interpretation superimposed on the physical environment [2].

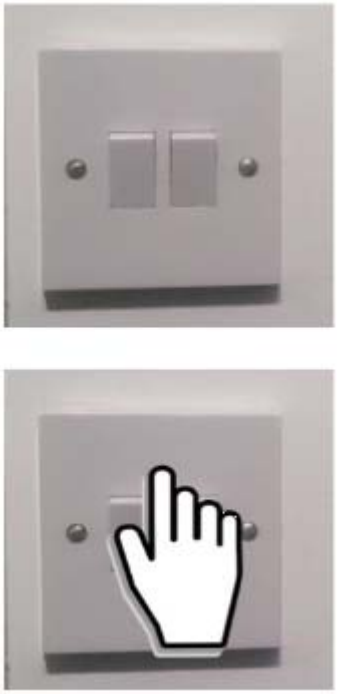

Figure 3: Use of augmented reality to make deictic references to actions attributed to physical objects.

In our augmented reality IoT programming interface, this allows the user to distinguish between an immediate “direct manipulation” action in the physical world (e.g. pressing a light switch) and reference to that action for the purpose of incorporating it into a program (e.g. referring to the use of that light switch at a time in the future). In this paper we have set out a theoretical account of multimodal deictic interaction with digitally-enabled physical devices, illustrated with many examples of past and future systems that demonstrate the advantages and disadvantages of alternative approaches. We believe that this account extends beyond a simple set of design rules, and represents a philosophy of end-user programming for the Internet of Things. The Internet is fundamentally a (very large) information structure. Although realised through engineering infrastructure, information structures are linguistic artefacts. The Internet of Things, however, is not solely linguistic, because physical “things” have an objective reality and embodied context that persist beyond the information they carry. In natural

language, the correspondence between information structures and physical contexts is maintained through deixis. However we choose to establish such correspondences for the Internet of Things, the result must also be deictic.

6 References

[1] S.Aghaee and C.Pautasso. 2014. End-User Development of Mashups with NaturalMash. J. Vis. Lang. and Computing, 25(4), 414-432.

[2] Z. Bai, A.F. Blackwell and G.Coulouris. 2012. Making Pretense Visible and Graspable: An augmented reality approach to promote pretend play. In Proc ISMAR '12, 267-268.

[3] A.F. Blackwell and R.Hague. 2001. AutoHAN: An Architecture for Programming the Home. In Proc. IEEE HCC'01, 150-157.

[4] R.A. Bolt. 1980. “Put-that-there”: Voice and gesture at the graphics interface. In Proc SIGGRAPH '80, 262-270.

[5] P.Dourish. 2004. What we talk about when we talk about context. Pers. Ubiq. Comput. 8(1)19-30.

[6] D. Edge and A.F. Blackwell.2006. Correlates of the cognitive dimensions for tangible user interface. J. Vis. Lang. and Computing, 17(4), 366-394

[7] J. H. Larkin. 1989. Display-based problem solving. In D. Klahr and K. Kotovsky Complex Information Processing: The Impact of Herbert A. Simon, 319-341.

[8] E. Toye, R. Sharp, A. Madhavapeddy, D. Scott, E. Upton and A. Blackwell. 2007. Interacting with mobile services: an evaluation of camera-phones and visual tags. Pers. Ubiq. Comput. 11(2), 97-106.