Human Computer Confluence

Alois Ferscha, Stefan Resmerita and Clemens Holzmann

Johannes Kepler University Linz
Institute of Pervasive Computing
Altenberger Straße 69, 4040 Linz, Austria
ferscha@soft.uni-linz.ac.at

Abstract. Pervasive Computing has postulated to invisibly integrate technology into everyday objects in such a way, that these objects turn into *smart things*. Not only a single object of this kind is supposed to represent the interface among the "physical world" of atoms and the "digital world" of bits, but a whole landscapes of them. The interaction among humans and such landscapes of technology rich artifacts happens to be more confluently, rather than on a per device basis. To address the confluence among humans and computing landscapes we study we study human gesticulation and the manipulation of graspable and movable everyday artifacts as a potentially effective means for the interaction with the physical environment. In detail, we consider *gestures* in the general sense of a movement or a state (posture) of the human body, as well as a movement or state of any physical object resulting from human manipulation. Further, based on the tangible user interface paradigm, we propose employing intuitive tangible universal controls that translate physical motions into actions for controlling landscapes of smart things. Such intuitive "everyday"-gestures have been collected in a series of user tests, yielding a catalogue of generic body and artifact gesture dynamics. We present a systematic approach to selecting and steering using tangible artifacts by associating a flip-movement to service selection and a turn-movement to parameter steering. An implementation of this approach in a general software framework and several experiments with various fully functional artifacts and devices are described.

1 Introduction

Computing devices are pervading already now into everyday objects, in such a way that users do not notice them anymore as separate entities. Appliances, tools, clothing, accessories, furniture, rooms, machinery, cars, buildings, roads, cities, even whole agricultural landscapes increasingly embody miniaturized and wireless – thus invisible – information and communication systems, establishing information technology rich socio-economic systems with the potential of radically changing the style of how we perceive, create, think, interact, behave and socialize as human beings, but also how we learn, work, cultivate, live, cure, age as individuals or in societal settings.

Prospective advances in microprocessor-, communication- and sensor/actuator-technologies envision a whole new era of computing systems, seamlessly and invisibly woven into the “fabric of everyday life” [22], and hence referred to as Pervasive Computing. Their services will be tailored to the person and their context of use. After the era of keyboard and screen interaction, a computer will be understood as secondary artifact, embedded and operating in the background, with its complete physical environment acting as interface (primary artifact). Pervasive Computing aims at interaction with digital information by manipulating physical real world artifacts as “graspable interfaces”, by simultaneously involving all human senses, and by considering interaction related to the semantics of the situation in which it occurs.

Future pervasive computing senses and controls the physical world via many sensors and actuators, respectively. Applications and services will therefore have to be greatly based on the notions of context and knowledge, will have to cope with highly dynamic environments and changing resources, and will need to evolve towards a more implicit and proactive interaction with humans. Communication must go beyond sending information from one fixed point to another, considering situations where devices cooperate and adapt spontaneously and autonomously in the absence of any centralized control. A vast manifold of small, embedded and mobile artifacts characterize the scenarios envisaged by Pervasive Computing. The challenges are related to (*i*) their ubiquity, (*ii*) their self-organization and interoperation, (*iii*) their ability of perceiving and interpreting their situation and consequently (*iv*) adapt the services they offer, the different modes of user interaction with those services.

All the considerations above show that the traditional understanding of having an “interface” among humans and computers vaporizes both from the interaction as well as from the technology viewpoint – considering human and computer activity at a confluence appears to be a more adequate characterization. In this paper, we discuss gestural interaction with everyday artifacts as a means for interacting with services provided by the surrounding environment. In contrast to traditional Human Computer Interaction, such interaction generally comprises ensembles of devices, where each device provides certain services. The user thus interacts with a *service composite*, which is composed of the single devices’ services, by using one or more graspable artifacts. As it does not make a difference from the user’s viewpoint if the used services are composed of multiple devices’ services or not, we just use the terms *device* and *service* without distinguishing from device ensembles and service composites henceforth, respectively.

2 Gestural Interaction

Human gesticulation as a modality of human-machine interaction has been widely studied in the field of Human-Computer Interaction. With the upcoming Pervasive and Ubiquitous Computing research field, the explicit interaction with computers with mouse, keyboard and screen in the WIMP metaphor has given way to a more implicit interaction involving all human senses. As an important part of this tendency, gestures and movements of the human body represent a natural and intuitive way to interact with physical objects in the environment.

Thus, manipulation of objects can be regarded as a means of intuitive interaction with the digital world. This paradigm underlies the research on Tangible User Interfaces (TUIs) [40]. Embodied interaction [29] [30] aims at facilitating remote control applications by providing natural and intuitive means of interaction, which are often more efficient and powerful compared with traditional interaction methods. TUIs couple physical representations (e.g. spatially manipulable physical artifacts) with digital representation (e.g. graphics and sounds), making bits directly manipulable and perceptible by people [31] [33]. In general, tangible interfaces are related to the use of physical artifacts as representations and controls for digital information [40].

We witness the advent of applications, appliances and machinery that are richer and richer in information technology, providing large palettes of services to end users. This richness brings up many challenges to the user interface designer, which must face the task of offering the user simple, natural, and intuitive interfaces to service providers, hereafter called *devices*. An important aspect of the user interface with devices is remote control, i.e., setting the inputs of services on devices. Most of today's devices come equipped with button-based remote controls. These are often badly designed, unnecessarily complicated, bound to specific devices, they tend to get misplaced, their usage is not intuitive and natural.

These issues can be addressed by the Tangible User Interface (TUI) paradigm [8], where button-based control artifacts are replaced with physical objects whose manipulation allows for intuitive and natural expression of control. In the sequel we present such an approach, based on the idea that device ensembles in physical space can be controlled by manipulating objects that reside in that space. We are interested in universal control, where the user is able to control multiple devices and device ensembles by using one or more artifacts. This tangible remote control paradigm provides an alternative to classical remote controls by using objects with multiple equilibrium states like cubes.

2.1 Discovery, Selection, Connection, Control

In order to control a certain device, the user needs to perform the following general sequence of operations:

1. **Device discovery**. Device discovery is necessary when the user is situated in a non-familiar space, as the user must know whether the desired device is available or not.
2. **Device selection**. The user must specify which device it needs to control. Alternatively, a certain device can be implicitly selected based on the user's context (i.e. information about his situation), preferences, and history.
3. **Connection**. The control artifact must be able to connect to the selected device. Thus, a communication channel must be established between the control artifact and the device such that control commands from the artifact can be relayed to the device.

4. **Device control**. A device offers a set of services, and the user manipulates the control artifact to set up input values for the services. To do so, the following steps are performed:
    (a) **Service discovery**. If the user is not already familiar with the device, then it needs to know the services provided by the device. In many cases, the user already knows which service it needs to control. For example, it is common knowledge that air conditioning devices have at least two services: temperature and fan power.
    (b) **Service selection**. The user chooses one of the services to control. For example, in the air conditioner case, it chooses temperature.
    (c) **Parameter steering**. The user sets up values for the controllable parameters of the service. For example, it sets a temperature value of 25^o C.

Considerable research and development efforts have been devoted to stage 1 ([19], [18], [21], [1], [20]), stage 2 ([19], [12], [9]), stage 3 ([1], [23], [24]), and stage 4a ([19], [1]). As for steps 4b and 4c, a combined approach for controlling the environment with physical artifacts, which allows to browse and select both devices and their services as well as to steer the input values of a selected service with simple gestures, is described in [6]. Without loss of generality, we consider only services with one parameter having a one-dimensional set of values. We assume that each device or service has a suitable user interface output which provides the user with appropriate feedback. Our approach is based on the simple but crucial observation that there exist two types of manipulations that can be intuitively associated to service selection and to steering, respectively.

In general, the geometry of objects suggests manipulation affordances in a TUI. The geometry of a physical object defines a number of stable mechanical equilibria of the object placed on a planar horizontal surface called a *flat*. A stable equilibrium of an object will be called an object *mode*. Once a control artifact is connected to a device, each service of the device is associated to a distinct mode of the object, under the assumption that the number of modes is greater than or equal to the number of services (a relaxation of this assumption will also be discussed). By a *flip-movement*, the user moves the object from one stable equilibrium to another, by changing the object's surface that touches the flat. Thus, a flip-manipulation triggers a change of the selected service. For example, a box has six stable equilibria. Hence, a box can be used to select from up to six services. A *turn-movement* is simply a rotation of an object. It is associated to steering the parameter value of the selected service. Figure 1 illustrates this assignment. Both movements are geometric rotations of objects. They can be executed with the objects in hand, without the need to place objects on actual surfaces. We shall later present experiments in a domestic setting, where objects of various shapes are employed for controlling a TV set and a music player. The artifacts are embedded with wireless orientation sensors to detect flipping and turning.

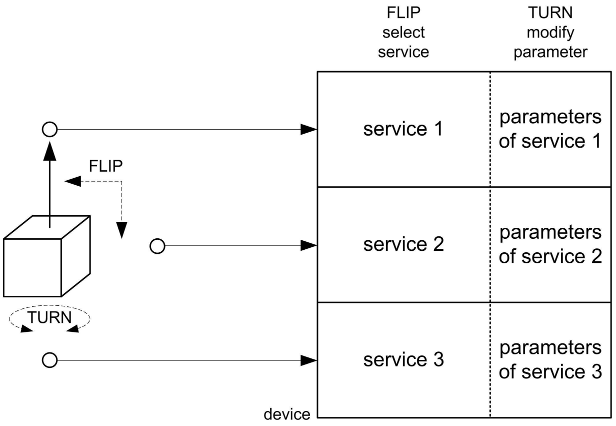


Fig. 1. The flipping and turning control actions.

2.2 Universal Interaction

Considerable research efforts have targeted the realization of a universal interaction device, which is a mobile computing system (usually a PDA or smartphone) that can be used for interacting with multiple services, e.g. the "Universal Information Appliance" [2], the "Universal Interactor" [5], and the "Personal Universal Controller" in [11]. The main issues are discovery of devices and services [19] and composition of user interfaces [14], [11]. Discovery is supported by service oriented frameworks, communication protocols and standards such as UPnP [18], Jini [21], Bluetooth [1], URC [20]. Proposed approaches for device selection include browsing, pointing and touching [19], [12], or automatic selection based on context clues and user history [9]. Connection is supported by wireless technologies such as Bluetooth, Zigbee [24], and WiFi [23]. In general, the universality of the control device means the ability to control multiple services, with as little a priori information as possible about the services. Such a handheld control device suffers from some of the shortcomings of today's remote controls: the device is complicated and therefore hard to use, it offers non-intuitive control means, it requires to be available at all times. In this paper, we consider several universality layers. In particular, more than one physical object can be used to control the same service. The redundancy of physical objects in the user's environment together with a dynamic mapping of objects and movements to services and parameters can ensure that a control object is always handy for any device that the user decides to control.

Tangible User Interface (TUI) research has studied the capabilities of physical objects as rich input devices. Specific movements of objects were considered for control. Tilting user interfaces [15] use the tilt of a portable device as input for the device. In [3], various artifacts and associated gestures are used for device control. Some works present objects where multiple faces are associated to different functions and flipping is used to select a function. In [4], flipbricks are described as part of graspable user interfaces. Different commands, such as "cut", "copy", "paste", are associated to each face of a flipbrick, and one of them can be activated by flipping the brick. The ToolStone device described in [16] uses also the rotation of the device, in addition to flipping, to further increase the selectable functionalities. Our paper builds on this research to investigate the use of such manipulations for remote control of devices. We do not assume particular shapes or types of objects and particular applications or services to be controlled. Thus, flipping and turning are considered here as manipulations of objects that can be generically mapped to abstract control actions.

An approach for using physical objects for home device control is reported in [10], where everyday objects and an augmented table are employed for configuring and using interfaces to applications. Objects must be placed on the table, and then they can be moved against the table. In contrast, we rely here on the geometry of the object to suggest the object's usage. As opposed to most of the papers described above, our implementation employs only orientation sensors for detecting the flip- and turn-movements (there is no sensing table or active surface on which control objects must be placed).

2.3 A Universal Remote Control

Tangible Interaction If the same artifact is used for both types of manipulation, then the turn-movement should be executed while the object is in a stable equilibrium state. This is guaranteed if the object is turned by an axis that is orthogonal to the related flat.

The association of flip- and turn-movements to service selection and respectively parameter steering is based upon the following observations:

- The service space of a device (or of an ensemble of devices) is a discrete set, with a relatively small number of services. The input space of a service can be a continuous set. On the other hand, the mode space of an object is also discrete with relatively few number of modes, while the space of the turning angle is a continuous set.
- The selected service should not change while the user is steering its parameters. This is why we are looking at stable equilibria. Thus, once a service is selected, the user is free to focus only on steering the value by turning.
- The stable equilibria can be used to disambiguate between intentional and unintentional control movements. To activate a service selection, the user must keep the artifact in a stable equilibrium for a certain amount of time (which can be heuristically determined at the design phase). If no mode is activated, then no control action is taken, regardless of the object's motion.

The artifacts that offer the flip and turn affordances can be everyday items or objects specially designed for control. Objects can have various numbers of modes, ranging from no mode (e.g. a ball), to one mode (a swivel chair), to tens of modes. (Notice that our approach is valid also when several artifacts are used to control a device at the same time. In this case, the modes of the artifact ensemble are given by all the possible combinations of modes of the component artifacts.)

The association of flip-movements to service selection and of turn-movements to steering is viable if the number of services is smaller than or equal to the number of modes of the involved artifact. If this is not the case, then a different association can be used, as follows: the turn-movement is associated to both service selection and steering, and the flip-movement is used to distinguish between the two cases. Thus, an artifact (or artifact ensemble) with at least two modes can fully deal with controlling large numbers of services.

Control types of a service For a given service, according to the set of all possible parameter values, we distinguish two control types: *discrete control* and *continuous control*. For example, the TV channel selection is a service with discrete control, whereas the TV volume is a continuous control.

There are two ways by which the user can set the value of a parameter: direct and sequential. In the *direct* type, the user is able to indicate the control value in one step, without browsing intermediate values. Examples are: setting a TV channel number with a classic remote control, setting a predefined value for room temperature. In the *sequential* type, the user browses input values until the desired value is reached. Usually, the sequential mode is required by the user-centric feedback-based control. Examples are: setting TV volume, setting light levels.

Universality requirements We seek remote control mechanisms that satisfy the following universality requirements:

1. The control artifacts should provide means for performing discrete, continuous, direct and sequential control.
2. The same device can be controlled by more than one artifact and different devices can be controlled by the same artifact. The mapping between control objects and devices should be dynamic. The user should be able to configure this mapping. In addition, the mapping can be context sensitive.
3. The mapping between artifact modes and services should be dynamic. This mapping can be context sensitive: the service that is associated to a mode can be determined ad-hoc, based on the user's context, rather than being a priori fixed or explicitly specified by the user.

Unintentional manipulations of artifacts (i.e., manipulations which are not targeted for explicit control) can be used for providing implicit control, whenever this supports the user's interaction with the device.

Implementation and Experiments We have employed the movements flip and turn for controlling devices in a domestic setting. Our experiments aimed at showing how the above universality requirements can be satisfied by our approach. In this respect, physical objects of various shapes have been embedded with orientation sensors that allowed a detection of the two manipulations. The devices to be controlled are a TV set and a music player. The experimental setup is depicted in Figure 2.

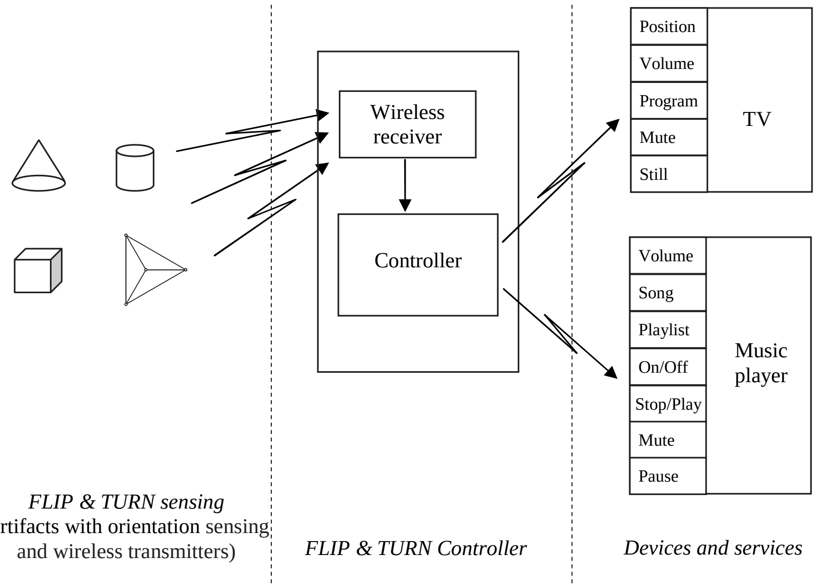


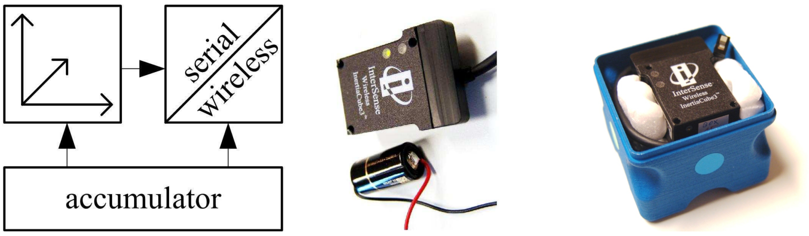
Fig. 2. The experimental setup.

Artifacts. We have used both everyday items and specially designed objects. Figure 3 presents some of the artifacts, which have no modes (e.g. a decorative orange), one mode (e.g. a swivel chair or a knob), two mods (e.g. a beverage can) and six modes (e.g. a cube).

(a) Orange (b) Swivel chair (c) Knob (d) Can (e) Cube

Fig. 3. Examples of control artifacts.

Sensors. We used InertiaCube3 wireless orientation sensors from Intersense [7]. The InertiaCube3 has 3 degrees of freedom (yaw, pitch and roll) with 360 degrees in all three axes. Together with a battery pack, the hardware is small enough to fit in half of a cigarette box. Figure 4 shows the embedded sensing.



(a) The orientation sensor (b) Embedded cube

Fig. 4. The embedded sensing.

Software. We developed a flexible software platform, which is able to accommodate a variable number of sensors and a variable number of applications. In this respect, we used Java and the OSCAR implementation of OSGi [13].

Devices and services. For rapid prototyping reasons, the two devices were implemented as computer applications. The TV is represented by a Java video player and Winamp is the music player. The TV screen and the Winamp graphical user interface are projected onto adjacent walls of the same room. The controlled services of the two devices are represented in Figure 2. The position service of the TV offers the user the possibility to change the horizontal positioning of the TV screen on the TV wall. Notice that each service has one parameter, which is also called a service *input*. There are services with continuous input space (e.g., volume, position), and with discrete input space (e.g., Program, Playlist, On/Off).

The following experiments now demonstrate various usage modes of flipping and turning for remote control, addressing the universality aspects described in the previous section.

Experiment A: The chair (Figure 3(b)) and the cube (Figure 3(e)) are employed to control the TV and the music player. The chair has one mode, and the cube has six modes. When connected to the TV, the chair's mode is mapped to the position service of the TV. Thus, a rotation of the chair steers the horizontal position of the TV screen on the wall such that the screen is always facing the chair. When connected to the Winamp, the chair mapped to the On/Off service: if the chair faces the Winamp wall, then the player is "On" (and the Winamp skin is visible on the wall). Otherwise, the player is "Off" (and the skin is not shown). The cube modes are mapped to the other services of the two devices. Thus, when the cube is connected to the TV, the user can flip the cube to change the service and then rotate the cube to steer the input. Snapshots from this experiment are presented in Figures 5(a) and 5(b).

(a) TV position control with chair

(b) TV program control with cube

Fig. 5. Experiment A. TV control with chair (one mode), and cube (six modes)

Since in this setting at most one device is used at a time, the chair is connected to both devices at the same time. When the chair is facing neither of the two walls or the user does not sit in the chair, both devices are off and hidden. If the user sits in the chair and the chair is oriented towards the TV wall, the TV screen appears on the wall at a position corresponding to the chair's orientation. Moreover, the cube is automatically connected to the TV device. Any subsequent rotation of the chair determines a steering of the screen's position on the wall. When the chair is steered to face the Winamp wall, the music player starts out and its skin is shown on the wall. The cube is automatically connected to the Winamp.

Let us consider now the universality requirements defined in the previous section.

1. Discrete control is provided by a suitable discretization of the rotation angle, and it is performed for example when changing programs on the TV, titles and playlists on the music player. For continuous control, the rotation angle

is directly mapped to the service input (possibly after a continuous transformation). For example, the distance between the left edge of the TV screen and the left edge of the TV wall is determined by a continuous nonlinear function $f : [a, b] \longrightarrow [0, L_w]$, where $[a, b] \subset [0, 360]$ is the interval of angles for which the chair faces the TV wall and L_w is the length of the wall. Sequential control is naturally achieved by using mappings from angles to the input space that are onto and monotone.

Direct control can be achieved by an unidimensional turn-movement only for two predefined input values (e.g., any clockwise rotation selects one value and any counterclockwise rotation selects the other value). More suitable for direct control is to flip the artifact, where predefined input values are defined as services and flipping is used to select one of them. Consider, for example, that a cube is used to control room temperature. By flipping the cube, the user can directly select one of six possible predefined values, while turning the cube can still be used for sequentially (and continuously) steering the temperature.

2. In this experiment, each of the two devices accepts control from the same artifact (e.g., the cube). Moreover, two artifacts are used to control the same device. The cube is dynamically connected to either the TV or the Winamp. The user determines this mapping by steering the chair. The use of context information for devices selection has been studied e.g. in [9].
3. The mapping of the chair mode is dependant upon whether the user sits in the chair or not. This context information regards the human motion activity. If the user does not sit in the chair, nothing happens when the chair is turned.

When the user sits in the chair and watches TV, chair rotations are usually not intended for establishing a certain position of the TV on the wall. They are rather triggered by factors which are external to the TV control problem (e.g., finding a more comfortable body position). Steering the TV screen such that it remains in the user's main attention is an example of implicit control.

Experiment B: A soft drink can (Figure 3(d)) is employed to control the music player. This illustrates how an object with only two modes can be used to control a device with many services by flipping and turning it. In normal position (i.e., with the top side up), turning the can selects one of the seven services offered by Winamp. To change from service selection to steering, the can is flipped. In upside down position, the can changes the input of the selected service. A snapshot of this experiment is given in Figure 6.

Experiment C: The chair (Figure 3(b)) and the knob (Figure 3(c)) are employed to control the TV. This shows how two single-mode objects can be used to control a device with multiple services. As in the previous experiment, one mode (of the chair) is used for service selection by turning and the other one (of the knob) is used for actual steering by turning. In this case, there is no flip-movement and the two turns can be executed in parallel.

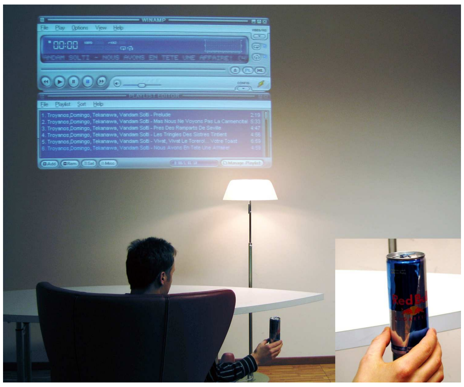


Fig. 6. Experiment B. TV control with soft drink can (two modes)

The experiments described above demonstrate the potential of tangible interaction for universal remote control. Moreover, they show how combinations of control artifacts can be employed to fully control fairly complex devices.

3 Conclusions and Future Work

The vision impacting the evolution of Pervasive Computing is the claim for an intuitive, unobtrusive and distraction free interaction with technology-rich environments. In an attempt to bring interaction *back to the real world* after an era of keyboard and screen interaction, computers are being understood as secondary artifacts, embedded and operating in the background, whereas the set of all physical objects present in the environment are understood as the primary artifacts, the *interface*. Instead of interacting with digital information via traditional computing means, Pervasive Computing aims at physical interaction with digital information, i.e. interaction by manipulating physical artifacts via *graspable interfaces*. It links the “atoms of the physical world” with the “bits of the digital world” in such a way, that physical artifacts are considered as being both representation of and control for digital information. Manipulating physical artifacts in the physical world hence causes the manipulation of their respective associations in the digital world and vice versa.

Motivated by the expressive power of gestures as enablers of intuitive interaction, we have presented a general approach to remote control of devices based on TUIs. Given that a remote control artifact is connected to a device, we proposed using a flip-movement for selecting the service of the device, and a turn-movement for steering the service’s input. This approach enables the achievement of universal remote control. The concept of universality considered in this paper has a larger scope than the usual case where a single artifact is able to control multiple devices. Thus, from our viewpoint, a universal remote control must be able to provide natural and intuitive means for continuous, discrete, sequential and direct control. Moreover, the same device can be controlled by more than one artifact and different devices can be controlled by the same artifact. The mapping from artifacts and manipulations to devices and services should be dynamic. This mapping can be determined in an ad-hoc manner, based on the user's context and preferences.

We showed how our approach can be applied to satisfy the above requirements by a series of experiments in a domestic setting. These experiments involved various objects of diverse shapes, including everyday items and specially designed artifacts. For the clarity of presentation, the unidimensional input case was considered. It should be noted that the turn-movement can be used to steer up to three dimensions. Thus, our approach is viable for most of the cases encountered in remote control of real world devices.

Clearly, to apply the proposed approach in fully mobile and ad-hoc settings, one needs suitable methods for device discovery and selection. In principle, on can employ existing solutions involving handheld devices (e.g., PDAs). Upon device selection, flipping and turning movements can be applied to the handheld object. Upon connecting a control artifact to a device, the mapping between the artifact modes and the services of the device must be automatically determined. We are addressing this issue at the device level. Thus, the artifact sends a description of itself to the device, which is then able to determine the artifact's modes. Challenging problems are how to use information about the user's context for achieving an intuitive mapping of modes to services, and how to make the user aware of this mapping.

With respected to the traditional understanding of considering an "interface" among humans and computers, it should have become clear, that it is rather the "confluence" among human activity and a sensitive, technology rich environment, that defines the "interaction" among humans and computers.

References

1. The Bluetooth specification. http://www.bluetooth.org.
2. K. F. Eustice, T. J. Lehman, A. Morales, M. C. Munson, S. Edlund, M. Guillen. A universal information appliance. *IBM Systems Journal*, Vol. 38, Nr. 4, pp. 575–601, 1999.
3. Orientation Sensing for Gesture-Based Interaction with Smart Artifacts. *Computer Communications*, vol. 28, no. 13, pp. 1552–1563, 2005.
4. G.W. Fitzmaurice. Graspable user interfaces. Ph.D thesis, University of Toronto, 1996.
5. T.D. Hodes, R.H. Katz, E. Servan-Schreiber, L. Rowe. Composable Ad-hoc Mobile Services for Universal Interaction. Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 1–12, 1997.
6. C. Holzmann, S. Resmerita, M. Leitner, A. Ferscha. A Paradigm for Orientation-Based Universal Remote Control. Proceedings of the International Workshop on the Tangible Space Initiative, in conjunction with Pervasive 2006, Dublin, Ireland, May 7, 2006.
7. The InertiaCube3 orientation sensor. http://www.isense.com/products/prec/ic3/wirelessic3.htm.

8. Ishii, H. and Ullmer B. Tangible Bits: Towards Seamless Interface to Access Digital Information. In Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '01), Seattle, Washington, USA, March 31 - April 5, ACM Press, pp.187–188, 2001.
9. K. Kaowthumrong, John Lebsack, R. Han. Automated Selection of the Active Device in Interactive Multi-Device Smart Spaces. Spontaneity Workshop at Ubicomp 2002.
10. C. Kray and M. Strohbach. Gesture-based Interface Reconfiguration. Workshop "AI in mobile systems" (AIMS 2003) at Ubicomp'03, Seattle, WA, USA.
11. B.A. Myers. Using handhelds for wireless remote control of PCs and appliances. *Interacting with Computers*, vol. 17, no. 3, pp. 251–264, 2005.
12. The Near Field Communication Forum. www.nfc-forum.org.
13. The OSGi Service Platform. http://www.osgi.org.
14. S.R. Ponnekanti, B. Lee, A. Fox, P. Hanrahan, T. Winograd. ICrafter : A Service Framework for Ubiquitous Computing Environments. Proceedings of Ubicomp 2001, LNCS 2201, pp. 56–75, 2001.
15. Jun Rekimoto. Tilting operations for small screen interfaces. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 1996), pp. 167–168, 1996.
16. Rekimoto, J. and E. Sciammarella. ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices. Proceedings of ACM User Interface Software and Technology (UIST), pp.109–117, 2000.
17. Ullmer, B., Ishii, H. and Glas, D. mediaBlocks: Physical Containers, Transports, and Controls for Online Media. In Proceedings of SIGGRAPH '98, Orlando, Florida USA, ACM Press, pp. 379–386, 1998.
18. Universal Plug and Play. http://www.upnp.org/.
19. P. Välkkynen, I. Korhonen, J. Plomp, T. Tuomisto, L. Cluitmans, H. Ailisto, H. Seppä. A user interaction paradigm for physical browsing and near-object control based on tags. Proceedings of the Physical Interaction Workshop at Mobile HCI Conference, Udine, Italy, 2003. http://www.medien.informatik.uni-muenchen.de/en/events/pi03/proceeding.htm.
20. G. Vanderheiden, G. Zimmermann, S. Trevin. Interface Sockets, Remote Consoles, and Natural Language Agents: A V2 URC Standards Whitepaper. http://myurc.com/whitepaper.php, 2005.
21. Jim Waldo. The Jini Architecture for Network-centric Computing. *Communications of the ACM*, July 1999, pp. 76–82.
22. Mark Weiser. The Computer for the 21st Century. Scientific American, vol. 265, no. 3, pp. 94–104, 1991.
23. The Wireless Fidelity Alliance. http://www.wi-fi.org.
24. The ZigBee Alliance. http://www.zigbee.org.
25. Amft O., Junker H., Tröster G.: Detection of Eating and Drinking Arm Gestures Using Inertial Body-Worn Sensors. Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC'05), pp. 160-163, 2005.
26. Bahlmann,C., Haasdonk, B., Burkhardt, H.: On-line Handwriting Recognition with Support Vector Machines - A Kernel Approach. In Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 49-54, 2002.
27. Benbasat, A.Y., Paradiso, J.A.: An Inertial Measurement Framework for Gesture Recognition and Applications. Gesture Workshop, LNAI 2298, pp.9-20, 2001.
28. Burges, C.J.C., A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2(2):1-47*, 1998.
29. Dourish, P.: Where the action is: the foundations of embodied interaction. MIT Press, Cambridge, 2001.

30. Fishkin, K., Moran, T., Harrison, B.: Embodied user interfaces: towards invisible user interfaces. Proceedings of the 7th international conference on engineering for human-computer interaction (EHCI'98), Heraklion, Crete, Greece, September 1998.
31. Fitzmaurice, G.W., Ishii, H., Buxton, W.: Bricks: laying the foundations for graspable user interfaces. Proceedings of the ACM conference on human factors in computing systems (CHI'95), Denver, Colorado, May 1995.
32. Greenberg, S., Fitchett, C.: Phidgets: easy development of physical interfaces through physical widgets. Proceedings of ACM symposium on user interface software and technology (UIST 2001), Orlando, FLorida, November 2001.
33. Holmquist, L.E., Redström, J., Ljungstrand, P.: Token-based access to digital information. Proceedings of the first international symposium on handheld and ubiquitous computing (HUC'99), Karlsruhe, Germany, September 1999.
34. Koleva, B., Benford, S., Hui Ng K., Rodden T.: A framework for tangible user interfaces. Proceedings of the real world user interfaces workshop at the 5th international symposium on human-computer interaction with mobile devices and services (MobileHCI 2003), Udine, Italy, September 2003.
35. Lementec, J.C., Bajcsy, P.: Recognition of Arm Gestures Using Multiple Orientation Sensors: Gesture Classification. 7th International IEEE Conference on Intelligent Transportation Systems, Washington, D.C., October 3-6 2004, pp 965-970.
36. Lenman, S., Bretzner, L., Thuresson, B.: Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction. Technical Report TRITA-NA-D0209, 2002, CID-172, Royal Institute of Technology, Sweden.
37. Chang, C.-C., Lin C.-J., LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
38. Nielsen, M., Störring, M., Moeslund, T.B., Granum, E.: A procedure for developing intuitive and ergonomic gesture interfaces for man-machine interaction. Technical report CVMT 03-01, 2003, Aalborg University.
39. Shimodaira,H., Noma, K., Nakai, M., Sagayama, S.: Dynamic Time-Alignment Kernel in Support Vector Machine. Advances in Neural Information Processing Systems 14, NIPS2001, 2:921-928, Dec 2001.
40. Ullmer, B., Ishii, H.: Emerging frameworks for tangible user interfaces. IBM Syst 39(3-4):915-931.
41. Westeyn, T., Brashear, H., Atrash, A., Starner, T.: Georgia Tech Gesture Toolkit: Supporting Experiments in gesture Recognition. Proceedings of ICMI 2003, November 5-7, Vancouver, British Columbia, Canada.
42. Williams A., Kabisch E., Dourish P.: From Interaction to Participation: Configuring Space Through Embodied Interaction. Proceedings of the Ubicomp 2005, LNCS 3660, pp.287-304, September 2005.