Using ARToolKit Markers to Build Tangible Prototypes and Simulate Other Technologies

Eva Hornecker[1,2] and Thomas Psik[1]

[1] Institute of Design & Assessment of Technology, Technical University Vienna,
Argentinierstr. 8, A-1040 Vienna
[2] Interact Lab, University of Sussex, Falmer, Brighton BN1 9QH, UK
eva@ehornecker.de
tomp@ims.tuwien.ac.at

Abstract. Quick prototyping of tangible user interfaces is currently hampered by availability of toolkits and the double challenge of tinkering with software and hardware. While software may be downloaded, hardware cannot. As a work-around for a class on experimental prototyping of tangible appliances we utilized the ARToolKit that tracks optical markers. By creatively adapting it, our students quickly developed working prototypes, simulating a range of devices and tracking technologies. Our approach enabled a focus on quick prototyping, idea testing and simulation of the interaction process. We explain our reasons for using the ARToolKit, summarize its advantages and disadvantages, present four students projects, and discuss our experiences and conclusions. In particular we found that visual tracking has the advantage not to limit or determine possible interaction styles and thus fosters designing richer interaction. We discuss this as a requirement for future tangible prototyping toolkits.

1 Introduction and Motivation

Quick prototyping of tangible user interfaces is currently hampered by the limited availability of toolkits and the double challenge of bricolaging with software and hardware. Sensing technologies are far from "plug and play" and require time to be mastered while each providing different constraints on what can be tracked [19, 23] and what style of interaction can be designed for. Wiring and soldering electronics requires a lot of time and competencies, which neither computer scientists nor designers usually possess [12]. Existing toolkits often consist of a combination of hardware and software [11,12, 10], only the software being available for free download. With limited budgets one is constrained in selection and often forced to decide on a specific technology too early. For teaching, these problems are even more salient. Such issues turned out as thresholds restricting proliferation of toolkits and accumulation of experience with TUI prototyping (especially for non-computer science communities) during a 2004 workshop about "Toolkit support for interaction in the physical world" [2].

For a class on experimental prototyping of tangible interfaces and appliances we used an existing toolkit widely used for Augmented Reality (short: AR), which relies on visual detection of optical markers ("glyphs"). By creatively adapting this toolkit – the ARToolkit [1], our students managed to quickly develop working prototypes of tangible interfaces, building a range of devices, despite of no budget and almost no

M.F. Costabile and F. Paternò (Eds.): INTERACT 2005, LNCS 3585, pp. 30 – 42, 2005.

hardware. Using optical markers and vision software, they simulated other kinds of sensing technology, which were not available. As this toolkit is easy to learn, stable and easy to integrate with other software, we avoided many technical problems.

Adapting this existing (and well working) toolkit from another domain and using it for tangible interaction prototypes provides an innovative work-around. Although the ARToolKit has previously been used to develop tangible interaction [8, 20, 21], it has to our knowledge up to now been rarely used *systematically* for "optical simulation" of tangible interaction technologies - in particular not in this variety - and has not been reflected as a teaching tool as well as in its virtues for quick prototyping and focusing on interaction design.

In this article we

- Describe the constraints we had to live with in doing this class
- Present the ARToolKit and the supplementary Open Tracker toolkit used
- Explain the advantages and disadvantages of using this vision technology
- Describe four student group projects, explain the simulated sensing technologies, compare which aspects of the product idea could be prototyped and experienced
- Describe experiences and lessons learned from this class.

1.1 The Class Held and the Constraints Motivating Our Choices

We gave a 3 hour class on "Experimental Design" (6 ECTS) in summer 2004 within the master program media informatics at the Vienna University of Technology. As teacher and practical support person were further involved Prof. Ina Wagner and Kresimir Matkovic. The bachelor program preceding this is technically oriented, providing students with little experience in iterative, user-centered, and creative design approaches. Other students may enter the master degree with other degrees, having less programming experience and different backgrounds.

Our aim was to introduce the students to experimental and creative prototyping methods (mock-ups, theatre and video prototypes) and have them iterate in idea generation and assessment. But we wanted to go beyond a design sketch, students should implement a working (rough) prototype as a proof of concept. With only 3 hours of lecture or presence time, this is – given the high load of classes required and the diversity of students – a wide scope. We needed time for introducing methods, idea generation and design reviews, leaving about a month for implementation. (usual for a technically oriented degree program like ours would be the opposite distribution). In addition we had no budget, could not buy hardware for students, and did not own much to lend away. Given our staff resources we needed to restrict students to a small range of technologies that we could give assistance for.

Having experience with barcode readers and the ARToolKit, we decided to restrict support to these technologies. Nevertheless we wanted the student groups to develop product ideas without feeling constrained by technology. They should focus on the product idea first, iterate and redesign it with consideration of the intended use context instead of being driven by technology. Therefore we introduced the available sensing technology only after the product ideas had been developed.

2 Using ARToolKit and Open Tracker

The main principle of the ARToolkit [1, 5, 15] is as follows. Visual markers (printable with a standard printer) are detected in a live video stream, extracting the 3D position of the marker (relative to camera position) and its rotation (relative to default orientation of the marker). We used the ARToolKit framework as basis, as the hardware needed as tracking device consists only of a web-cam. The markers have to have a certain look and the size of the markers depends highly on the camera resolution being used. It is a well known and often used framework in the AR-community, this ensures that the framework is thoroughly tested and stable. Furthermore we used the Open Tracker library (developed at Vienna University of Technology, IMS Institute [13, 17]) that delivers an abstracted data stream from the tracking device. For our students we provided a compiled version of the software, which reduces the installation process to copying the files. We also included a ready-to-use configuration file for the server, thus reducing the setup procedure to a minimum.

Open Tracker [13, 17] provides an abstraction layer for tracking devices. Support for a number of tracking devices and also other tracking frameworks are included in the library. The library is well documented and is being used as a basis for the AR system Studierstube [13, 14,17]. The library includes a network sink that sends tracking data to the network. The tracking server is configurable through a XML-file, where the tracking devices and the sinks can be defined. The framework also allows transforming the data, before it is sent to the network. This however requires advanced understanding of 3D coordinate system manipulation and calculation. The output is a stream of tracking data including, besides position and orientation, a quality measure of the data. It does not provide support for interpretation of the data or event handling, like “marker appeared” or “marker removed”. The detection of these events has to be implemented in the application layer. In order to ease this generic task we provided the students with a small Java class as a template for their own implementation. This already performed some basic functions like reading data from the Open Tracker network stream and producing events.

In [15, 5] the usage of the ARToolKit markers and their restrictions as well as details on the toolkit itself are described in detail, focusing on the application area of augmented reality. Other contexts of usage of the same library are presented in [14]. The ARToolKit has been used to develop several tangible interfaces [8, 20, 21]. One TUI toolkit resembling the ARToolKits capabilities in many respects is Papier-Mâché [16], which integrates different tracking technologies besides vision and provides higher support for programming and debugging.

The tracking technology chosen influences what can be tracked and therefore interpreted (cp. [23, 4]): (a) presence of (unidentified) objects, (b) identity of objects, (c) position, (d) orientation, (e) movement (discrete or continuous) (f) relations of objects to each other (g) activation and other events (besides of movement). „Real“ image recognition being slow and difficult, image recognition is usually restricted to attached barcodes or optical markers. Problems result from occlusion through hands, body or other objects, delaying the system reaction until markers are visible again. Further problems are: stability, robustness, and especially changes in light [19, 4, 24], requiring close control of lighting conditions. Clever choice of markers like in the ARToolKit and size of markers improve reliability and speed of recognition. Unfortunately barcodes and markers from a users viewpoint are distracting, not task-related and not aesthetic (see [24]). Besides of barcodes [16] some systems rely on markers reflecting ultra-violet light [6]. We will now reflect on advantages and disadvantages of these ARToolKit markers in the context of prototyping tangible interfaces and appliances.

2.1 Advantages and Disadvantages of the ARToolKit in the Tangibles Context

We were not aware of advantages beyond easy learnability, fast tracking, and price when we decided to use the ARToolKit. We became aware of some of the advantages reported below only in reflecting upon the class and the different student projects. The same holds for disadvantages.

Advantages

When comparing the ARToolKit markers with other kinds of sensing technology, we find several advantages. Unlike most RFID systems, detection is not restricted to adjacent surfaces and very precise. We can detect absence, presence, position and orientation of tags (the last is (almost) impossible with only one RFID). In principle, one can track markers in 3D space, only limited by visibility of the marker within line of sight. Detection is fast, allowing for tracking movement and for simultaneous presence of several markers. In addition there are no cables necessary, which do restrict interaction with some 3D tracking devices. Markers can be attached anywhere and need not be built into objects.

For these reasons prototyping with optical markers allows for a wide range of movements and styles of interaction. The toolkit itself does not restrict interpretation of events to simple event-effect mappings, thus allowing for more sophisticated interaction patterns. This allows pushing tangible interaction design beyond imitating GUI interaction styles (the ubiquitous button pushing and slider shoving) and inventing more varied interaction styles which take account of human bodily skills, are expressive in movement and make interaction enjoyable [7, 9]. Tracking in free space furthermore allows to go beyond shoving objects around on flat surfaces (the dominant interaction style of most existing TUIs). Some larger toolkits like Phidgets [11, 12] which integrate several types of sensors, that allow similar freedom use accelerometers, which can be deployed in three axis which give continuous outputs ss do the force and light sensors

With optical markers we have almost no hardware costs. The software includes a module for printing out new markers and mapping them to an ID. As a video camera most web-cams suffice. The software is highly reliable, being in wide use within the AR community and developed in cooperative effort by several research groups worldwide. By providing a compiled version of the software and a ready-to-use configuration file, we could reduce the installation and setup procedure to a minimum. Unlike many other hardware tracking toolkits, where calibration often takes hours and is vulnerable to many factors (electro fields, metal, water, other materials....) calibration is easy and quickly done, supported by specific software.

In effect our students got going with the ARToolKit in a day or two, being able to concentrate on implementing their idea concept, instead of indulging in the idiosyncratic problems of sensing hardware. Using the optical markers and vision technology, the groups were able to test the core functionality of and the interaction with their product. As will become salient in the presentation and discussion of students' working prototypes, testing the feel of interaction did work for many areas.

2.2 Disadvantages of Using Optical Markers and Vision Tracking

A well known problem of vision tracking is the control of lighting [19]. Changing levels of light and limited contrasts disable correct registration, similar to the problem of the angle between light and camera. The ARToolKit requires relatively large black surfaces, which printed out with some laser printers tend to reflect the light, giving highlights in the video image. A better solution is to use ink-jet printer, or to adjust the angle between light, markers and camera. Related is the problem of tag occlusion. Tags need to be fully within camera view to be detected. Thus occlusion by the interactors' bodies or by stacking objects makes tagged objects virtually absent. If one marker overlaps with another marker, the overlapped one will not be detected. Furthermore the camera field determines the interaction space and limits it. Additionally markers in 3D are only registered as long as they are visible within a certain angle of orientation (one cannot turn them around 180º).

Marker tags need to be visible and thus may interfere with aesthetics and intuitive legibility of objects. The looks of a tangible interface, "simulated" with tags, often differ from the intended product and may distract users and evaluators from the general idea. The required tag size (for detection) also limits the smallness of registered objects. And over time tags will deter, fade or get dirty, thus endangering long-term usage of tags (this is less of a problem when prototyping) [24, 19].

For some goal technologies one may need to invent a clever set-up – an example of how this can be done is presented later-on. Nevertheless there are limits to what kind of sensing technology and product idea is simple to prototype and simulate. While for large devices optical markers and the camera may be hidden inside the device, this is not possible for small devices (e.g. a handheld with many buttons). For these a prototyping toolkit such as CALDER [18] will be more appropriate. As another example, a device that controls lighting would disable its own tracking conditions.

A disadvantage of using the ARToolKit is that it only eases the registration process and the creation of events. Different from tangible prototyping toolkits there is no easy mechanism for connecting events with resulting actions. Interpretation of events and output of interaction (system response) must be implemented in standard programming languages, requiring some programming experience. Necessary is also programming of basic position calculations. As it is used mostly for Augmented Reality, the toolkits eases detection of markers and overlaying an image at the appropriate point in an AR display. When designing tangible interfaces and appliances, there may be different requirements on programming support, better served by toolkits designed specifically for TUI prototyping, as [16, 3, 11, 12].

3 Student Projects

We present four out of six student projects here. These were the best (either in original idea or in iterating and implementing it) and do suffice for showing the diversity of sensing technology simulated. Two of these are very innovative in simulating mechanical or electrical sensing respectively GPS, the others are variants of common ARToolKit uses. As common theme the class was given “home and household”. Some students had attended a previous class on investigation methods, these were allowed to stay with this topic (tourists in Vienna). In total 28 students took part in the course. We describe the basic product idea and the working prototype, focusing on which aspects of the final product (look, handling, interaction process) the prototype helped to experience and assess.

The Mimic Music Maker

This group focused on the selection of music titles from a database. The title selection should be based on the users mood and personal preferences. The user should also be able to enlarge and refine the database on the fly. To enforce the emotional character of the device, the group decided to use a mask as interface, what also gave it a playful aspect. The final device would have the form of a full head instead of only a mask. For identifying oneself (choosing settings) one would put a hat or something else on top (with tracked RFIDs) to identify oneself. The mask should have a well visible switch (with legible state) for selecting the “set track mood” mode. Manipulating its facial expression defines the mood (happy, sad, angry …) the currently playing track is connected with in the database. The group implemented the main functionality of choosing a music style by manipulating the masks facial expression.

This device would be implemented using potentiometers or other kinds of electrical sensing for registering manipulations. Students resorted to mechanical engineering, making use of the fact that a mask has a big backside to hide the mechanics and put up a camera behind. One could move the eyebrows and the mouth to form a smile, a neutral look, or a frown. Levers and sliders manipulated on the face are connected with mechanics behind it and move optical markers. The camera tracks this movement. The group was able to prototype the look of the device and to test a good amount of its interaction feel (restricted by some problems in building the mechanics).

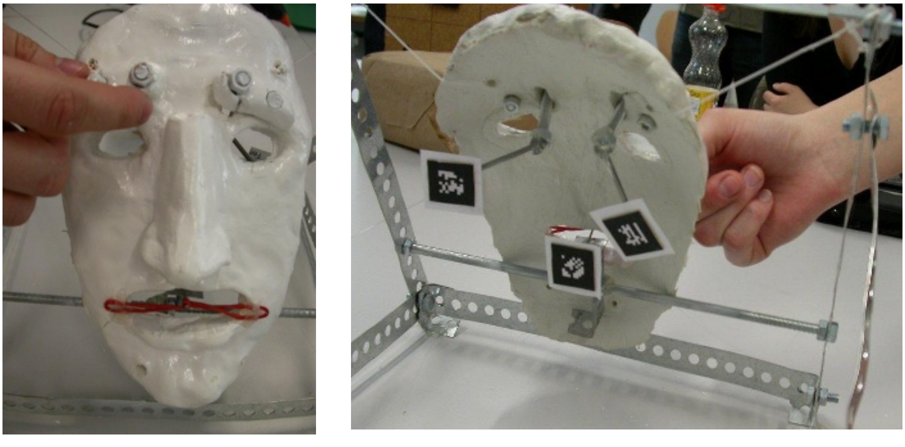

Fig. 1. Mimic Music Maker: Manipulating the mask and optical markers moving at the backside, visible for the web cam positioned behind the mask

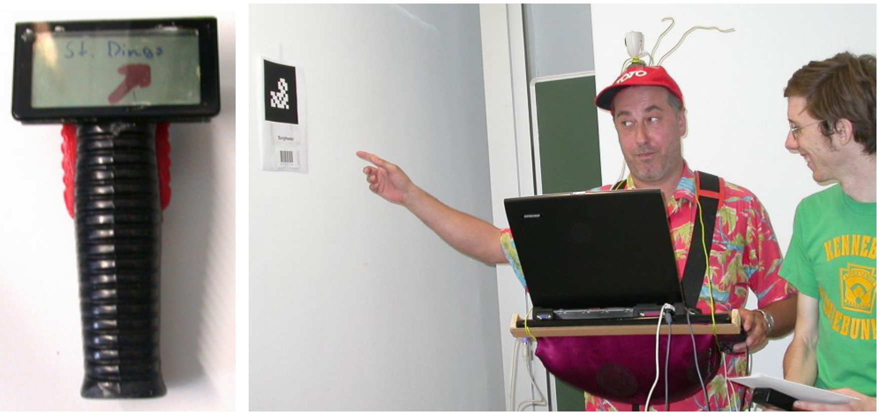

Fig. 2. Tourist Navigation Device: Mock-Up with display and vibrating (red) parts on the sides. Demo of working prototype - the system detects a visual marker (camera worn on hat) of a site and spoken text output is triggered.

With some bricolaging many other mechanically manipulated devices could be emulated in a similar way, using levers, strings etc. that move optical markers somewhere on the backside of the device or in a distance. Not for all kinds of devices this will be as easy. Levers and strings must be attached somewhere and optical tags put so they do not interfere with interaction. This limits the smallness of devices that can be designed. A negative effect of using vision detection is that the position of the camera must be fixed relative to markers. The mechanical construction of this combination – movable markers and fixed camera position – is one of the major problems for students not educated in mechanical construction.

Tourist Navigation Device

This group had in the previous semester undertaken an ethnographic study on tourists in Vienna. Building upon this experience, they developed the idea of a device that enables finding interesting sites while walking serendipitously through the city. The device would tell the tourist strolling through the city if (s)he comes close to anything previously marked as interesting so (s)he does not walk past. The device could also enable following a given path, if switching into guided mode. This group initially developed a video prototype showing the use of two versions for their product in the inner district of Vienna (performed and role-played by group members). This in-situ experience helped them in deciding upon form factors (see mock-up in figure 2 left) and interaction style for the device.

The product idea consists of selling the device along with city guides marked with optical codes. City visitors use the device to scan codes for those sites they want to see. If walking through the city, the device vibrates if coming close to a site. The small display would show where to go and the name of the site. This could be complemented with the appropriate sides of the device vibrating (left, right, both sides). Information boards at attractions could be augmented with optical markers. On scanning these, the tourist would hear explanations via earphones.

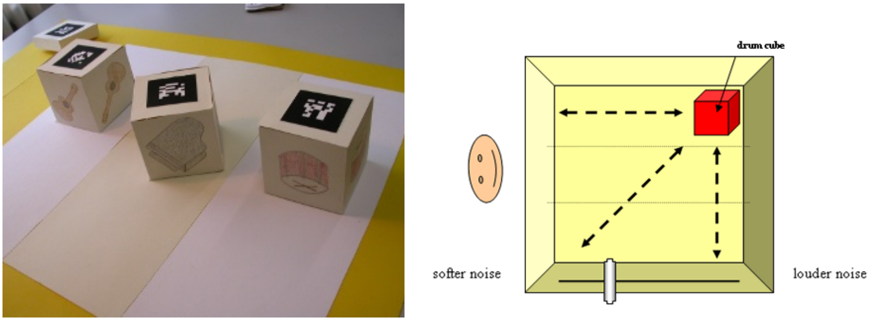

Fig. 3. Composing Cubes: Blocks and playing board. Principle of working.

The goal technology for realizing this device would be GPS (or cell phone cells for identifying location) plus orientation sensors and a bar code reader. The group employed optical simulation of location detection by wearing a camera on the head and strategically placing optical markers in the room. This way they could simulate the interaction process that a tourist would experience and explore potential interaction patterns and problems. This supports iterative software-development. A positive side effect is that testing the software and simulating interaction can occur anywhere, independent of "real" locations by just hanging up markers. For the working prototype, the computer had to be carried around by the test person, as the camera needs to be tethered to it. Therefore the looks of the device did not resemble the design idea at all and the concrete feel of interaction, especially of manual handling, could not be simulated. But the student group had spent a lot of time on deciding on form factors, tinkering a non-functional mock-up well in advance.

Composing Cubes

This group iterated their idea several times, starting from the (too complicated) idea of a puzzle for blind and sighted people with musical feedback which also allows composing music. They eventually decided to focus on a music composing device (c.p. [8, 20]. The system consists of a playing board with different areas. Different cubes represent different musical instruments. The playing board is divided into three visible columns from left to right of the player. These columns represent three different effects (echo, reverb…). Moving cubes on a column controls volume. On the right-hand side a slide-sensor can be moved up and down. This regulates the tempo. For a more advanced version it was envisioned to use three cameras set up at 45° angles from the board, recognizing e.g. stacking of cubes. Turning the cubes over and setting it back on the board activates a different melody.

For such a system one can imagine using either vision or field sensing as implementation technology. E.g. AudioPad (building up on SensePad) uses RFID tracking [19]. Most prototypical systems use optical markers [8]. As students demonstrated, response times are good enough and the system works well under stable lighting conditions. With vision tracking, further forms of manipulation are possible, such as occluding a cube to stop a track from playing. Vision tracking suffers mostly from the big markers on top of elements, making it difficult to place intuitive icons on them

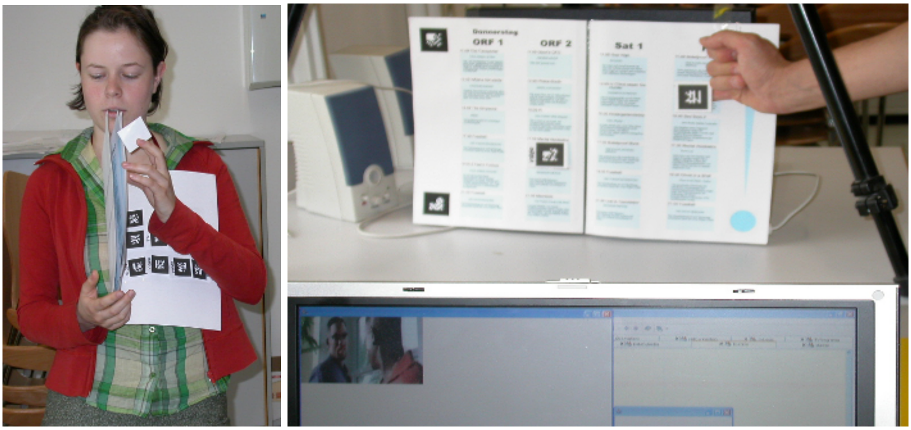

Fig. 4. Interactive TV: The TV magazine has a reservoir of adhesive optical markers. Demo of attaching markers to shows (VCR programming) and selecting a show (starts the TV and selects channel (demoed on computer).

and to make them aesthetically pleasing. Due to stable and quick registration, the prototype system provided a close experience of the interaction feel that a system with field sensing would have.

Interactive TV

The product idea of this group was a TV magazine that enables controlling the TV set and programming the VCR from the magazine. Neither would one need to search the remote control nor remember which channel is placed on what number or how to program the VCR. This group started out in developing this product idea by role-playing situations in a theatre-like way.

The magazine must be placed on a location where it is visible for the camera. Pointing with the finger (or a pointing tool) to any TV show starts the TV and selects the channel, if the show/movie is currently running. Attaching post-its with optical markers to a TV show programs the VCR to record it. Attaching another kind of marker switches the TV and the channel on as soon as the selected show starts. The magazine has a supply of markers on its last page. An advantage of these markers is their persistence, giving an overview of what a family wants to record or see in a week. Browsing through the TV magazine would remain as usual. Deciding upon what to record could take place anywhere, as the magazine is moveable. Zapping would still be easy to do by pointing. Here the goal technology would be vision, albeit probably using infra-red markers, so that visual icons can be legible for laypersons.

The prototype enabled simulating potential looks of such a system, the feel of using it and experiencing the interaction process (albeit without a real TV, using a computer to simulate responses). The prototype served well as a proof of concept. An advantage of using the ARToolKit that here got salient was the possibility of 3D interaction, when selecting shows by pointing.

4 Our Experience with This Class and Lessons Learned

Our students highly enjoyed this class and its experimental character. Although the time given for implementation was considered by most as too short, they would not have missed the time for idea generation and iteration and the exposure to creative prototyping methods. They were proud of their ability to bricolage and to invent work-arounds or tricks in simulating non-available sensing technologies. Getting the tracking working and implementing the product idea in the last month of the semester, which is crowded with hand-ins and exams, challenged students a lot. Student feedback taught us, that despite of the early focus on methods we should give out the ARToolKit framework earlier (done in the new run of the class). This would allow time to experiment with set-up of the system, registration, and calibration. Unfortunately it may also interfere with having groups develop ideas freely (without having in mind technical constraints).

All groups had chosen to focus on home entertainment from the common theme of „home and housekeeping". As we could see from the tourist project group, detailed (ethnographic) investigation of a theme has high impact on the product idea, improving contextual knowledge and awareness of factors affecting use of the device. Most groups were not as aware of factors on usability, desirability or practicability of devices. Yet considering the limited time available, we are happy with the results.

Students programmed in Java, Perl and C^{++}, sometimes using several computers for different aspects of the functionality. Some of the groups needed little support for programming, others needed support in design principles and methods. With basic programming experience the functionality itself was usually easy to implement, as there were no complicated algorithms included in the project ideas. Therefore team building should ensure a sufficient range of competencies within groups. We observed that most groups developed a division of labor with some members responsible for design and physical tinkering and others concentrating on programming. For our class with its focus on process this is fine. If everybody should acquire experience in programming or in visual design, additional exercises and lectures would be necessary and there might be less focus on inspiring and creative design methods. In the new run of the class more emphasis was put on students declaring their competencies and assignment of responsibilities for e.g. documentation, interaction design (responsible for facilitation, approaches of prototyping), programming, project coordination and media design (documents, visuals, sound, video….).

For students with a basic computer science education and some proficiency in programming it was easy to get the ARToolKit working and to develop simple applications using its data input. For students with different backgrounds the challenge is much higher, especially as for some applications geometric calculations for position and orientation are necessary. In standard AR applications the marker is simply visually overlaid in the display with an image. For our purposes, tracking events must go through further processing, extracting appearance, disappearance or movement of markers. As second step these events are mapped with resulting actions. Here a toolkit such as Papier-Mâché [16], or a graphical mapping of events and actions such as in iStuff [3] might be beneficial in lowering thresholds for non-programmers.

The simple technology used for implementation enabled an (uncommon) focus on the design process and idea or scenario generation. Except for one group all were able to present a working prototype showing core functionality. Most groups had started out generating ideas with no particular tracking technology in mind and did manage to implement these. The examples given demonstrate that it was possible to simulate a wide range of tracking technologies and to prototype various kinds of devices.

On reflecting the resulting prototypes another advantage of using optical markers became obvious. The technology does not limit or determine possible interaction styles. One can move markers around freely – or alternatively the camera – resulting in a continuous flow of events. Interaction thus is not restricted to pushing buttons or touching specific sensorized points. Movement can be in 3D and simultaneous. The type of events interpreted thus can differ widely. Effort is moved towards the algorithms making sense of detection events. Such an algorithm may e.g. create meta-events depending on previous events. As indicated earlier, creating such kind of software requires more programming experience.

5 Conclusions and Summary

A major problem of tangible toolkits is that only the software can be downloaded via internet – hardware parts with sensors and actuators must be bought, configured or self-soldered. Tinkering with electronics requires a lot of time (even for people who do this more often) and competence in fields, that computer science and design students and practitioners are not well trained in. The specific quality of Tangibles – to be tangible and physically embodied– renders sharing (of tools, results, systems) more difficult in these respects. Our approach provides a work-around for teachers and researchers, which do not have the resources to buy or develop their own technologies, but want to focus on quick prototyping and idea testing.

Using optical markers and the AR toolkit enabled our students to quickly prototype tangible interfaces while not prematurely closing down the idea space. Student groups invented optical simulations for different tracking technologies and device types. Our choice was originally mostly due to our constraints concerning funding, available hardware and the kind of support we could give to students. Observing the results of student project work revealed additional advantages. Interaction styles are not limited to button pushing and sliders or to shoving objects around on a table. The toolkit allows for interpretation of a continuous flow of events, which can also be simultaneous. Interpretation is not restricted to simple mappings of discrete events with one-click-effects. On the other hand, effort is shifted towards the algorithms interpreting the dataflow, raising demands on programming experience.

Our assumption that the toolkit would be easy to set-up, use and to integrate with other software proved correct – at least for the kind of students we had in this class. We assume that student groups without members having programming experience will experience more problems. A remedy might consist of additional toolkit modules, which enable easy mapping of events with actions, e.g. by graphically connecting event types with actions and devices, as in iStuff [3]. The Phidget toolkit [11, 12] enables mapping incoming physical events with button clicks of standard GUI applications. One such system the Equip Component Toolkit (ECT) [25] allows designers to engage with constructed system components through a GUI that displays the flow of information instigated by a particular action, as it occurs. Nevertheless such mechanisms often tend to predefine what constitutes an event, doing some filtering of events and defining possible kinds of mapping.

Djajadiningrat et al [9] recommend emphasizing the expressiveness of interaction, especially in bodily interaction – that is the how of acting affects the effect. But most toolkits do not support value ranges, combining several inputs, continuous action-event couplings (besides of discrete, button-pushing like events). Hardware toolkits also can limit expressiveness, if they restrict interaction to pushing and sliding – one could also rub, move, hit or stroke a button. Ingenious designers will be able to nevertheless design innovative and expressive forms of interaction. But what is easy to do will be used by those who are less inventive, have less patience or do not know better. Toolkits may, by making it easy to develop an exact, event-based language of interaction, discourage exploring the richness of interaction meaning and style.

We do not consider development of tangible prototyping toolkits to be unnecessary, on the contrary. Yet there is currently only a handful of such toolkits and few research teams developing them. Given that there is only a limited number of people investing time in developing this software, progress is still slow. The AR toolkit on the other hand is being developed as an open source project with lots of people from the Augmented Reality community contributing to it.

We do not claim to be the first using the ARToolKit for developing tangible interfaces/systems. But we (respectively our students) seem to be among the first to use it explicitly for quick prototyping of tangibles and to emulate/simulate such a wide range of different tracking technologies. Most publications focus on one system designed or on the toolkit itself. This paper laid focus on interaction design when analyzing our students' prototypes and discussed toolkits in terms of what style of interaction they lend themselves to design for. Growing experience in using tangible prototyping toolkits and comparing experiences with different toolkits will advance the community in understanding strengths and weaknesses of toolkits and in setting up requirements for future toolkits.

Acknowledgements

Thanks to all participants of the Pervasive 2004 Workshop on Toolkit Support for Interaction in the Physical World, the students participating in our class, Prof. Ina Wagner for teaching the class with us and Krejomir Matcovic for helping us in supporting the students in working with the ARToolKit. Eric Harris from Interact Lab provided comments on technical issues of toolkits and sensor.

References

1. ARToolKit Download Page http://www.hitl.washington.edu/research/shared_space/download/
2. Ballagas, R., Klemmer, S., Sheridan, J.: Toolkit Support for Interaction in the Physical World. Workshop at Pervasive'04. 20.4.2004, Vienna. (2004) http://media.informatik.rwth-aachen.de/tuit/

3. Ballagas, R., Ringel, M., Stone, M., Borchers, J.: iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments. Proc. of CHI'03. ACM (2003). pp.537-544
4. Benford, S. et al: Sensible, sensable and desirable: a framework for designing physical interfaces. Tech Report Equator-03-003 (2003)
5. Billinghurst, M., Kato, H., Poupyrev, I.:The MagicBook: A Transitional AR Interface. In: Computers and Graphics, November (2001). pp. 745-753.
6. Brown, L.D., Gao, C., Hua, H.: Toward a Tangible Interface for Multi-Modal Interior Design using SCAPE. Proc. of IEEE VR'04 Workshop Beyond Wand and Glove Based Interaction. (2004). 79-83
7. Buur, J., Jensen, M.V., Djajadiningrat, T.: Hands-only scenarios and video action walls: novel methods for tangible user interaction design. Proc. of DIS'04. ACM (2004) pp. 185-192
8. Constanza. E., Shelley, S.B., Robinson, J.A.: D-TOUCH: A Consumer-Grade Tangible Interface Module and Musical Applications. Proc. of HCI 2003. Bath, UK (2003)
9. Djajadiningrat, T., Overbeeke, K., Wensveen, S.: But how, Donald, tell us how? On the creation of meaning in interaction design through feedforward and inherent feedback. Proc. of DIS'02. ACM (2002). 285-291
10. Gellersen, H., Kortuem, G., Schmidt, A., Beigl, M.: Physical Prototyping with Smart-Its. In IEEE Pervasive Computing 3(3) (2004). pp. 10-18
11. Greenberg, S., Boyle, M.: Interaction in the real world: Customizable Physical Interfaces for Interacting with Conventional Applications. Proc. of UIST'02. ACM (2002). 31-40
12. Greenberg, S., Fitchett, C.: Phidgets: Easy Development of Physical Interfaces through Physical Widgets. Proc. of UIST'01. ACM (2001). 209-218
13. IMS ARToolKit Page http://www.ims.tuwien.ac.at/~thomas/artoolkit.php
14. Kalkusch, et al: Structured visual markers for indoor pathfinding. Proc. of IEEE 1st Int. Workshop on Augmented Reality ToolKit, (2002). technical report TR-188-2-2002-13, Vienna University of Technology.
15. Kato, H., Billinghurst, M., et al.:Virtual Object Manipulation on a Table-Top AR Environment. In: Proceedings of International Symposium on Augmented Reality ISAR 2000, (2000). 111-119
16. Klemmer, S.R., Li, J., Lin, J., Landay, J.A.: Papier-Mâché: Toolkit Support for Tangible Input. Proc. of CHI'04. ACM (2004). pp. 399–406.
17. Lee, J.L, et al: The Calder Toolkit: Wired and Wireless Components for Rapidly Prototyping Interactive Devices. Proc. of DIS'04. ACM (2004). 167-175
18. OpenTracker Download Page http://www.studierstube.org/opentracker/index.html
19. Patten, J., H. Ishii, J. Hines & G. Pangaro (2001). Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. Proc. of CHI'01. ACM (2001) 253-260.
20. Poupyrev, I.: Augmented Grooove: Collaborative Jamming in Augmented Reality. Proc. of SIGGraph'00 Conference Abstracts. ACM (2000). 77
21. Sinclair, P., et al.: Links in the palm of your hand: tangible hypermedia using augmented reality. Proc. of Hypertext and Hypermedia. ACM (2002) 127-113
22. 22.Ullmer, B., Ishii, H.: Emerging Frameworks for Tangible User Interfaces. IBM Systems Journal 39(3 & 4) (2000). pp 915-931.
23. Ullmer, B.: Tangible Interfaces for Manipulating Aggregates of Digital Information. Dissertation, Massachusetts Institute of Technology (2002)
24. Want, R., Fishkin, K.P., Gujar, A., Harrison, B.L.: Bridging Physical and Virtual Worlds with Electronic Tags. Proc. of CHI'99. ACM (1999). 370-377
25. ECT http://ubisys.cs.uiuc.edu/proceedings_04/toolkit_rapid_construction.pdf