Cognitive Dimensions of Tangible Programming Languages

In *Proceedings of the First Joint Conference of EASE and PPIG* 2003, pp. 391-405

Alan Blackwell
Computer Laboratory
University of Cambridge
afb21@cl.cam.ac.uk

Abstract

We have in the past described several approaches to programming with tangible interfaces – both complete programming languages implemented in physical form, and simpler physical interfaces (for example the front panels of domestic appliances) that have some programmable capabilities. In this paper, we explore the more generic possibilities of tangible interfaces as a basis for programmable functionality. Our specific target is the development of a query language that can be used in school classrooms, based on physical elements that contain radio frequency ID (RFID) tags. This research project is still in progress, but an important early stage has been a theoretical analysis, based on the Cognitive Dimensions of notations (CDs) framework, that has provided guidance regarding what kinds of programming might be feasible using such an interface. This is a novel use of the CDs framework, insofar as it does not aim to evaluate a specific kind of interface, but rather investigates what kinds of interface we might imagine making. We believe this to be a valuable perspective for the design of non-WIMP interfaces and other ubiquitous computing innovations.

1. Introduction

There has been a steady trickle of proposals for programming languages where the basic elements of the language are not displayed on a screen, but are actual physical objects. The AlgoBlocks system (Suzuki & Kato 1995) is a paradigmatic example. In AlgoBlocks, lexical elements of the LOGO language were assigned to sealed metal boxes, about 20 cm on each side. These could be assembled by plugging each block, via connectors on the sides, into neighbouring blocks. A complete LOGO program could be constructed by assembling a sufficient number of blocks on a tabletop.

The physical assembly approach to computer interfaces that was applied in AlgoBlocks had earlier been pioneered by researchers such as Aish (1979), who built a system of intelligent modelling bricks that could communicate with each other and with a host computer to describe their own arrangement in three dimensions, and thus be used as a basis for 3D solid modelling. Physical or "tangible" interfaces such as AlgoBlocks and Aish's system are now becoming quite familiar through the prolific work of Ishii's Tangible Media Group at MIT (e.g. Lee et. al. 2000, Patten et. al. 2000), and in a variety of initiatives (such as the European "disappearing computer" programme, Weiser's "ubiquitous computing", and AT&T Research "sentient computing") that integrate computational interactions into the physical environment.

1.1 Defining Tangible User Interfaces

These Tangible User Interfaces (TUIs) are mainly distinguished by the fact that they are an alternative (or supplement) to conventional display and input devices such as CRT/LCD monitors, keyboards, mice, styli, touchpads, trackballs etc. Conventional user interfaces are mainly designed for text entry and display, supplemented by 2D graphics and a pointer control device. Although the keyboard and pointer control devices of conventional user interfaces are physical objects, they are not considered TUIs. TUIs must also be distinguished from the virtual reality paradigm, where imaginary objects are rendered in 3D, possible "immersive" (filling the whole field of view) and controlled by sensing the position of the user's body, head, or hands. TUIs, in contrast to either of these conventional alternatives, employ actual physical objects that can be grasped, positioned and manipulated in the real world. A variety of sensing devices can be used to detect the position of these objects, and feed their configuration back to a host computer.

Few TUIs aspire to the status of AlgoBlocks as a complete programming language. Nevertheless, there seems to be an intuitive appeal around the idea of building a computer program out of something like Lego bricks (Resnick et. al. 1996). Unfortunately, there are substantial technical challenges and expense involved in creating a large number of intelligent components with connectors that are both physically and electrically robust. As a result the current programmable Lego product (Mindstorms) simply puts a single microprocessor into a single brick, rather than composing a programme out of multiple bricks, and the programming is done on screen then downloaded, rather than assembled from Lego bricks (it is not, in fact, a TUI). In contrast, our Media Cubes system for domestic programming (Blackwell and Hague 2001) is one example of a programming language that is built solely out of tangible elements, as was the case with AlgoBlocks.

1.2 The WEBKIT Project

Based on earlier reports of our work on Media Cubes, we were invited to join WEBKIT, a European research consortium with a very specific emphasis on physical user interfaces for querying the world-wide web. The WEBKIT proposal had already specified an application area (school classrooms) and a communication technology (radio frequency identification, or RFID) that would be used to implement the interface. As these decisions had been taken before we joined the project, our first action was to carry out an analysis of the interaction possibilities that these constraints might allow. The actual functionality of the TUI had not been determined, although the proposal assumed some form of query language. The application of the query language was unclear – it might be used purely for information retrieval, for scripting, for data collection or presentation of authored material. Our theoretical analysis was intended to identify any design constraints that would influence this decision on grounds of feasibility.

This paper reports the results of that analysis. It is instructive not only as design research relevant to tangible user interfaces, but also as an example of a design that is being carried out from first principles, based on Cognitive Dimensions (CDs) analysis (Green 1989, Green & Petre 1996, Blackwell & Green 2003).

2. The Analytic Framework

The transition from graphical user interfaces to tangible interfaces, like the historical transition from textual to graphical interfaces, is motivated by concerns of usability. The intention is that tangible user interfaces (TUIs) should make human interaction with a computer more efficient or effective for some purposes, just as GUIs did. However, much of the development of GUIs has actually proceeded without direct consideration of relevant psychological research, relying instead on popular intuitions such as “a picture is worth a thousand words”. This tendency in GUI research has been criticised at length (Blackwell 1996), and encourages unsafe assumptions that can be disproven experimentally (Green, Petre & Bellamy 1991, Blackwell 2001).

The same tendency is now apparent in recent research into TUIs. As in earlier GUI research publications, claims regarding the cognitive benefits of tangible interface are often based on loosely defined and vague ideals such as “natural and intuitive” (e.g. Koike et. al. 2000, Lertsithichai & Seegmiller 2001) rather than empirically supported cognitive theory. Similar ideals motivated the WEBKIT project, whose title “intuitive physical interface to the world-wide web” is phrased as an assumption, rather than a research question.

In this paper we consider the cognitive properties of TUIs from different perspectives, corresponding in broad terms to the linguistic analysis levels of lexicon, syntax and semantics. At the lexical level, the cognitive properties that are relevant to TUIs relate to their status as manipulable indexes and mnemonic cues. At the syntactic/semantic level, the TUI constitutes a notational convention for a physical query language that can be explored using Cognitive Dimensions.

At each level, we use the terms “textual”, “graphical” and “tangible” for the convenience of researchers familiar with these terms in the computer science context. They do not have a simple one-to-one correspondence with cognitive processes. Reading a book, for example, employs linguistic processes, processes of visual perception, and physical manipulation skills. Even “textual” computer systems share these characteristics, so the cognitive properties of TUIs must be explored without assuming any necessary difference between human use of text, graphics and tangible interfaces.

2.1 Lexical Level: Tangible Symbols

Writing systems are distinguished according to whether they are alphabetic (using more-or-less phonetic spelling) or ideographic (using a symbol to represent a concept). The lexical elements of graphical user interfaces combine ideographic elements (icons) with alphabetic elements (labels). This provides a number of benefits to the user:

- some visually distinctive icons can be located easily during visual search (the “pop-out” effect)
- associating an image with a verbal stimulus can enhance memory (the “dual-coding” effect)
- presenting alternatives on the screen can improve memory retrieval (“recognition versus recall”)

However the use of picture-plus-image codes in GUIs also brings some disadvantages:

- They require more screen area, so users must scroll to see them
- graphic vocabulary conventions are less well developed than verbal language, so icons other than literal pictures may have reduced neural activation levels.

These advantages and disadvantages are likely to be exaggerated in tangible user interfaces

- It is likely that manipulation of physical objects will further enhance mnemonic effects by multiple encoding
- But 3D arrangements of tangible items reduces retrieval performance with respect to 2D arrangements (Cockburn & McKenzie 2002)
- Reaching to grasp a close physical object can be faster than moving a mouse pointer
- But moving to grasp a physical object that is out of reach is slower than scrolling a GUI window
- We have relatively few conventional physical symbols, other than the conventions of literal scale representation used in physical models.

The potential difficulties in using a tangible lexicon were anticipated over 200 years ago by Swift in Gulliver's Travels. The foolish professors of Lagado claim "that since Words are only Names for Things, it would be more convenient for all Men to carry about them, such Things as were necessary to express the particular Business they are to discourse on." (Swift 1726, part 3, chapter 5). The idea that people ought to carry huge collections of objects in order to save effort in talking is no more sensible than the idea that TUIs are inherently superior to GUIs.

These observations from previous research lead us to conclude that the use of physical symbols in TUIs will provide few benefits at the lexical encoding or retrieval level. There may be a slight mnemonic advantage resulting from further modalities of encoding, but these are likely to be compromised by the reduction in efficiency and exploitation of symbolic conventions that results from the constraints of physical icons. At the lexical level, we conclude that TUIs will not be "natural" or "intuitive" as intended for the WEBKIT project – these advantages must be sought at the syntactic, or semantic levels.

2.2 Syntactic/Semantic Level: Tangible Notational Systems

In spoken language, most syntactic structure is carried through "one dimensional" arrangement of the lexical items into a temporally ordered sequence. The placement of lexical items within this ordered context carries meaning, so both syntax and semantics are determined by the ordering relations *before* and *after*. The ways in which we can use spoken language are restricted by the structural ground of syntax – one word must appear either after or before another, and it is not possible modify the start of a spoken sentence once the speaker has moved on (words cannot be "un-said").

Textual computer languages are also interpreted by the computer as a simple sequence, although use of special layout characters within that sequence (such as linefeed to mark the end of a syntactic unit) are seen by human readers as introducing a second dimension of arrangement on the page (the relations left, right, up and down). Furthermore, computer languages are entered using editors that allow changes to be made to earlier parts of the sequence (unlike spoken languages, text on a screen can be "un-said"). These factors allow human writers to add extra information, for example by lining up corresponding pieces of text underneath each other. When considered as a notational system, this flexibility allows new possibilities for syntactic structure and corresponding semantic richness.

Graphical languages provide for a far greater range of relationships between their elements, including connection, intersection, topological inclusion, superimposition and others. Different graphical languages choose to exploit some of these graphical variables (Bertin 1981) for syntactic purposes, but not necessarily all of them. A well-designed graphical notational system provides editors and syntactic/semantic conventions that are easily used by human writers (e.g. not constraining the order in which the notation is constructed, or making it too difficult to change) and readers (e.g. exploiting the visual relationships between the elements to express consistent meanings).

The possible applications of graphical notations have been thoroughly analysed from a variety of perspectives, including Bertin's graphical semiotics (Bertin 1981) as well as Green's CDs. Some of these analyses are applicable to tangible notation systems, as discussed below. In general, Bertin's analysis forms a foundation for CDs, insofar as it specifies the form of the notational medium.

3. "Notational Variables" of Tangible Notations

Bertin, in a framework since updated by MacEachren (1995) and Engelhardt (2002), analysed the information-carrying properties of graphical displays in terms of the conventions of ink placement on a page. Variously called "graphical variables" or "variables of the plane", they include size of marks, their orientation, colour, texture etc. in addition to their position within a 2D coordinate frame. Each of these variables can be made to correspond to a dimension of information, with some restricted to expressing categorical or ordinal relations rather than the ratiometric relations of a continuous positional dimension.

The domain of physical, tangible, objects is, of course, far more complex than the domain of ink distribution. Every sensible property of the physical world could be used to express information, and it is hard to predict which of these may be exploited in tangible interfaces of the future. Guided by Bertin and his successors, this section attempts an initial survey by adaptation of that structure.

3.1 Positional Variables

The most obvious distinction between the two dimensional plane and the three dimensional physical world is the increased number of *positional variab*les in the physical world. If physical objects are placed on a flat surface, they can be used in all the ways defined for the plane. In addition to use of the plane, physical objects can be placed at different elevations above the flat surface. This is however a restricted dimension as (unlike virtual reality representations), objects can only be placed above the surface if resting on another object. This is often characterized as offering 2.5 dimensions rather than 3 dimensions of positional freedom. The restriction is analogous to the restriction on spoken language – it is possible to add another word after one that has already been spoken, but it is not possible to add a word before one that has already been spoken. Interestingly, it is also possible to achieve the same effect to some extent in a 2D representation by partially superimposing multiple marks. Gestalt principles of boundary perception allow us to recognize a small number of such superimposed marks.

3.2 Orientational Variables

Orientation of physical objects also allows, at first sight, a greater number of degrees of freedom, in that physical objects can be rotated on three different axes. However objects will not rest in arbitrary orientations on three different axes. They can be rotated in place when resting on a flat surface (carrying one continuous dimension of information), and can be balanced on any flat side (carrying a categorical dimension of information) with the restrictions that a) the user must be able to identify the side, and b) this may interfere with rotational position. Identification is not trivial – if an identification mark is placed on the flat side concerned, this will not be visible when the object rests on that side. This can effectively limit the number of available sides to shapes where the human viewer sees a direct correspondence between the side that it rests on and the side that is labeled (cubes and some prisms, but not solids with larger numbers of sides at non-orthogonal angles such as icosahedra or dodecahedra). Rotational position must be encoded by the human viewer in terms of reference points on the object itself, and also a frame of reference within the environment. The reference points on the object, furthermore, must be available for unambiguous interpretation regardless of which side the object is resting on. This restriction has meant that many tangible user interfaces simply use flat elements with only two possible resting positions – one side or the other side.

3.3 Temporal Variables

The *temporal* behaviour of physical objects is very different to that of graphical elements. The position of graphical marks on paper is persistent, whereas the position of physical objects can be easily modified (unless glued in place). Computer displays allow graphical marks to be moved by the user, and also to be animated by the computer for a variety of purposes, including restoration of the display to its state before the user modified it. Tangible displays can be moved by the user, but cannot usually be animated by the computer. This means that previous states of the display cannot be restored automatically, and also that tangible interfaces are generally input-only devices. These temporal characteristics mean that tangible interfaces can either express a single configuration, created interactively before "freezing" by transfer to the computer, or else a series of configurations, each of which is transient from the user's perspective, though they may be stored in the computer. This use of a temporal sequence to represent sequence is a natural mapping, but has severe disadvantages, as noted in the Cognitive Dimensions analysis below.

3.4 Structural Variables

The shape of physical objects allows for a wide range of *structural variables*. Physical objects can be articulated so that parts of the object move relative to each other. They can also employ fastenings so that parts can be added or removed. There are hundreds of such mechanical elements, analysed in frameworks such as Reuleaux's collection of kinematic pairs (1876). The level of complexity at which a physical object ceases to be a reconfigurable structural token and becomes a mechanism is arguable. For the purposes of tangible interfaces, the information carrying potential is probably best defined by the availability of digital encoding devices such as rotary encoders, strain gauges, gravitational accelerometers and so on.

3.5 Material Variables

The material properties of physical tokens can also be used to carry information. These could include density, heat conductivity, surface friction, rigidity, surface hardness and many others, as well as more familiar sensory codes such as colour, odour etc. Most material properties are a manufactured feature of a physical object, and cannot easily be altered by users. The space of material properties is also not fully orthogonal (it is difficult to find materials that are both flexible and have high surface hardness), limiting the freedom with which they can encode abstract information.

3.6 Surface Markings

In addition to all the properties listed above, physical objects can also have markings placed on their surface. In most cases such markings are identical to those we consider as textual or graphical representations - they are not part of the physical object, and have a psychological status independent of the surface on which they are imposed (Ittelson 1996). This is, of course, a desirable attribute of many information-carrying objects. Paper is valuable precisely because it facilitates the use of surface markings with minimal attention to the physical properties of the object. Previous research projects into tangible interfaces have often found that it is important that users be allowed to make markings on the surface of the tangible tokens. This is discussed further under the heading of secondary notation below.

3.7 Design Implications

Physical objects provide a rich set of information-carrying variables. Many of these can be used for relational purposes, as would be required in constructing a physical programming or query language as required in WEBKIT. Structural variables are particularly well-suited to carrying relational information. Orientation can be used to express a small number of categories. Exploitation of these variables is limited mainly by the Cognitive Dimensions of Notation use (discussed in the next section), and by the technical limitations of RFID tags (discussed in the section after that).

4. Cognitive Dimensions of Tangible Notations

This section assumes that the reader is familiar with the Cognitive Dimensions of Notations framework. Research based on the framework has been published many times at this conference in the past. A recent overview of the framework (with some recent modifications) can be found in Blackwell and Green (2003), or Blackwell, Britton et. al. (2001). The discussion in this paper does include some terms that have been added to the framework subsequently to better-known early publications such as (Green 1989, Green & Petre 1996).

4.1 Medium, Marks and Environment

Tangible user interfaces fall into the class of “transient” notation, in that users do not create permanent marks. The medium on which the notation is “written” is the medium of space, and “marks” on that medium result from the presence of an object at a particular spatial location. A collection of objects arranged in a particular way can thus carry meaning in the same way as a collection of marks made with a pencil on a piece of paper. Moving any one of the objects changes the meaning, hence the transience of the notation when compared to paper and pencil, where the pencil can be removed while leaving the marks. In a more complex transient notation, marks can also be located relative to each other in time as well as space. This can be regarded as a kind of manual animation, in which the motion of objects, their removal and addition, all carry meaning. Inspecting the current state of the objects will reveal only the end product - the meaning of the notational message can only be revealed by replaying the sequence of actions. In this case, the “marks” of the notation consist not just of objects, but of events that are located in space and time.

Although the arrangement of physical objects, when used as a notational system, is transient, physical objects can be used as an interface to create a less transient notation - for example one that is displayed on a computer screen in response to the user’s manipulation of the objects. In this case, the notation on the screen is a new notational layer, which can be analysed separately from the transient arrangement of the physical objects. The notation as it is displayed on the screen is not a tangible notation in the sense discussed here. It may be a graphical notation, in which case the contribution of the physical interface can be assessed in terms of whether it makes the manipulation of the graphical notation more or less difficult than (say) using a mouse to manipulate it.

4.2 Activities

In the classroom context assumed for the WEBKIT project, construction of queries or other scripted interaction is often likely to involve *transcription*, as a teacher demonstrates a program that should be constructed, or students try for themselves programs that have been provided in books.

Students attempting new programs are most likely to do this in a lesson context where they have access to a record of previous programs that can be used as a template. This will involve transcription followed immediately by *modification* to adapt the previous example to the new case.

Neither the languages nor the classroom context are likely to require regular *incrementation* of previous programs, other than in the specific teaching strategy where students are provided with purposely incomplete programs. In this case, the required completion has been anticipated by a teacher, so will be adapted to the notational environment available.

Search within a specific program is trivial, as the whole program is likely to have only a few elements. For the same reason, there is little opportunity for *exploratory design*, as programs are restricted to a relatively small number of overall structures, and these are of limited complexity.

4.3 Visibility

Transient notations have inherent problems of visibility. If the notation uses space and time as its medium, a message produced in that notation can only be viewed by viewing a movie of the marking events. If that movie is paused, the visible display is only a temporal "slice" out of the sequence. It is very difficult to compare two alternatives (juxtaposability), because the two movies must either be displayed simultaneously (with the user's attention divided) or viewed one after another (with the user remembering all the details in the first while comparing them to the second).

If a tangible interface is used to create a stored expression, the stored version will be invisible as soon as the tangible objects are moved. It is not normally practical to have the computer move the tangible objects in order to make visible some stored configuration. The usual solution to this is to introduce another notational layer that can be viewed on the screen, thereby overcoming the visibility problems of the tangible notation layer.

4.4 Diffuseness

A common criticism of graphical notations is that they are diffuse – they take up a lot of screen real estate to convey an equivalent quantity of information to text. This problem is exacerbated in the case of tangible notations, which take up even more space in three dimensions. Tokens can be made smaller, but are then difficult to manipulate. The need to choose a fixed size, either large or small, illustrates another challenge of tangible notations – the scale cannot be zoomed, as physical objects stay the same size. An advantage over graphical displays is that it is relatively easy to see detail in context, by attending to local features within a field of view not limited by the size of a screen as physical objects can be placed all around a room.

4.5 Viscosity

It is easy to modify arrangements of physical objects in space, especially where there are no structural constraints connecting them. It is far more difficult to modify a sequence of events in time. Once the sequence has been demonstrated, it is very hard to make a small change to that sequence without travelling back in time. The only option is to demonstrate the whole sequence again, resulting in extremely high viscosity.

4.6 Secondary Notation

Spatial arrangement and grouping of tangible elements can provide a valuable channel either for personal prompting, or to illustrate related teaching objectives. Spatial secondary notation will be transient in the case of a tangible interface. An important opportunity for persistent secondary notation is the ability to annotate tangible elements by sticking things onto them, writing on them, or otherwise changing their appearance in ways that will not change their system status.

4.7 Hidden Dependencies

It is possible to indicate fixed relationships between tangible elements through morphological similarity, but these require interpretation and search by the user. Transient dependencies can only be suggested by physical proximity, and even this does not indicate specific properties of the relationship. Some tangible interfaces have used wiring panels to indicate relationships with physical connections (Patten, Griffith & Ishii 2000), but these are subject to tangling, making it difficult for users to trace the connections. There is no convenient tangible equivalent to the ubiquitous node and link formalism that is used to indicate relationships between independent graphical elements in 2D. As a result TUIs seem more likely to suffer from hidden dependencies than GUIs.

4.8 Role Expressiveness

The true power of tangible notations lies in their rich correspondence to physical objects used in other contexts. This provides many opportunities to model elements of the notation on familiar objects, and thereby exploit the associations of those objects to express the role of that element within the notation. A great deal of emphasis has been placed in WEBKIT on using objects that are already relevant to the classroom situation.

4.9 Premature Commitment

As with visibility, this is a serious problem in the case where the notation includes a dynamic element. If the order in which the user constructs the query notation is itself a part of the notation, then the user has no choice with regard to the order in which a specific query might be created. This is a particularly serious form of premature commitment, although it might be acceptable if the only activity required is transcription (because a given query can then be constructed in order from the example given). If the notation only involves arrangements of objects in space, rather than ordering of events in time, this problem will not arise to the same extent.

4.10 Progressive Evaluation

If a tangible interface is used as a manipulation interface for a graphical display, then the display can give continual feedback regarding the status of the expression being constructed. If the tangible notation involves any arrangement of objects that must be complete before it can be interpreted, the separation between the physical objects and computational domain may make it difficult for users to monitor the results of their work so far. This separation does not occur in a GUI.

4.11 Provisionality

Various strategies are available for provisional arrangement of tangible notations. Objects might be arranged outside the context in which their positions are sensed, or with the computer interface turned off. As physical objects are always manipulable, there is no problem in "playing around" while thinking about potential arrangements. TUIs could well be superior to GUIs in this respect.

4.12 Abstraction

If new functions are assigned to tangible elements, there are quite severe physical limitations on the types of abstraction that can be expressed. Constant values can be assigned to physical tokens, but variables and functions are less straightforward. As objects can only be present in one place at a time, it can be difficult to represent parameter binding, and separating definition from invocation results in the problems with hidden dependencies and visibility already noted. This is more severe than for GUIs, where function invocation can be expressed by using some identifier or visual token that relates the invocation site to the definition site. Those identifiers are copies of some part of the actual declaration, and as noted previously, it is less straightforward to copy elements of a TUI than of a GUI.

4.13 Design Implications

Relying on a temporal sequence of object placements results in several serious usability problems that limit the complexity of tasks that can be undertaken with tangible notation elements. Tangible elements also provide some notational advantages, but these are most likely to be realized in cases where sequence of object placement is not the primary syntactic medium. The alternatives are to distinguish physical positioning of elements, and to allow element configurations to persist beyond the interactional context. In any case, it is essential that users be provided with some means to annotate the resulting elements or configurations for their own use.

5. Notational Potential of RFID Tags

The previous sections have described the potential usability benefits of tangible interfaces on the basis of human manipulation and perception, without concentrating on which manipulation styles can be sensed by using RFID tags, as chosen in the WEBKIT project. This section reviews the sources of information that can be used to define the syntactic structure of a TUI, and relates them to this technology. It is the interaction between these technical constraints and the notational concerns discussed in the previous sections that finally determine the envelope of possible designs for our language.

RFID devices have previously been used in a variety of TUI prototypes, and they bring significant advantages to WEBKIT. They were chosen because they are small and cheap (a few pence each), so can be used economically in the classroom situation, where unattached components of teaching hardware are quickly lost. They include no power source (being powered by induction when brought close to the RFID antenna), so need not include batteries – recharging or replacement of worn batteries is another major source of difficulty when classroom equipment must be maintained by teachers.

5.1 Detection of Proximity

RFID aerials are tuned to detect the presence of tags at a specific distance (or detection envelope) from the aerial – but the detection envelope is a fixed property of the reader, not modifiable under computer control. This means that the actual distance of a tag from the aerial cannot be determined, only the fact that it is either within the detection envelope or outside it (with some zone of uncertainty about the envelope perimeter).

The orientation of a tag relative to the reader cannot be determined. It is possible that some orientations may reduce electromagnetic coupling so that it appears the tag has moved outside the detection envelope. This would have the additional consequence that orientation cannot be used for purposes of secondary notation (as cues to the user).

Multiple readers having relatively small, non-overlapping detection envelopes can be placed in an array in order to detect relative position of tags (to a resolution determined by the number of aerials). There is no other means by which an RFID reader can determine spatial position of a tag, beyond the fact that the tag has or has not been detected within its envelope.

5.2 Detection of Tag Identity

When responding to the detection envelope, all tags transmit a unique ID. If tangible interaction elements are grouped into classes, it will be necessary to establish the correspondence between a given tag ID and the class of notational elements to which it belongs.

5.3 Detection of Tag State

Some tags include memory, allowing information to be written to the tag, and echoed back to a reader. This tag state is not, of course, visible to the user. Tag state is only relevant if the tag is being used to carry information between computers – if the tag was used only with a single computer, the state of that tag could be stored in the computer, and indexed by the tag ID. Tag state introduces the possibility that the history of a user's interaction with that tag can be associated with the tag.

5.4 Detection of Sequence

Typical RFID interfaces are able to detect presence of a tag within a period of one to two seconds (although this may vary for low frequency and high frequency type tags). This is too slow for accurate judgment of rhythm to communicate additional channels of information (such as double clicks). As tags have unique identifiers, the order with which they are moved in and out of the envelope can be detected, especially if they are moved with one hand. If one tag is held in each hand, and they are moved toward the reader in rapid succession, it is likely that the reader may misread a sequence that appears quite unambiguous to the user. This means that if sequence is used as an element of the notation, it will be necessary to take some care over situations in which the user might work with both hands.

5.5 Detection of Relations between Tags

Syntactic formalisms require that there be some kind of relations between the lexical elements. RFID readers are actually quite limited in the range of relations between tags that can be detected. Low frequency tags allow the reader to uniquely identify all tags that are within the detection envelope at the same time. Using this technology, users can employ a notation in which the syntax is based on simultaneous presence of any two (or more) tags are present on the reader at the same time.

This is not the case with high frequency tags, where readers are only able to identify a single tag ID at a given time. In this case, the only way for a user to establish a relationship between tags is to place them within the aerial envelope one after another (a temporal notation). This results in all the disadvantages of temporal relations discussed above.

If multiple aerials are used to detect location of the tag, spatial relations can be used to support a far more sophisticated notation. As the spatial coordinates would not be continuous, the user would have to be guided by marked locations at which tangible elements can be placed. Those marks could give additional cues with regard to required syntax – in fact they could employ physical cues to assist the user in creating valid expressions.

5.6 Detection of Structural Configuration

The general analysis of tangible interfaces given earlier noted several advantages that would arise from the use of elements that are mechanically articulated to allow different structural configurations. In linguistic terms, this can be considered either as a lexical morphology (analogous to verb conjugation), or as a closely coupled syntactic construct supporting special relationships between the component parts. As we have noted, the local detection of such configuration variables is mainly dependent on suitable transducers and encoders. However this assumes that a connected processor is available to interrogate the state of those transducers. In the case of RFID tags, the tag itself has very limited processing power. Most tags simply report an identification code, while more sophisticated tags can be made to store a code transmitted to them, and report it back later. RFID tags are not normally able to request state information from local peripheral sensors. It would be possible to construct an intelligent device that used the RFID transmission protocols to report locally sensed state information, but the design of such a device would be challenging, especially if it had no local power supply. For these reasons, the detection of structural configuration in the TUI appears to be beyond the scope of the WEBKIT project.

6. Conclusion

In this paper, we have investigated the potential of tangible interfaces as a basis for programmable and abstract functionality. The specific target of our work is the development of a query language that can be used in school classrooms, based on physical elements that contain radio frequency ID (RFID) tags. This research project is still in progress, but an important early stage has been a theoretical analysis, based on the Cognitive Dimensions of Notations framework, that has provided guidance regarding what kinds of programming might be feasible using such an interface.

This is a new way of using the Cognitive Dimensions framework. It is unusual for a design exercise to be quite so loosely constrained as this, because most user interfaces are constructed to suit established hardware standards (at first "1D" text-stream teletypes, now "2D" bitmapped displays with pointing devices). The move to "3D" TUIs is at present relatively open to innovation, but as there are not many more dimensions to come after the third, it is possible that this style of Cognitive Dimensions analysis will not often be required in the future.

However the analysis conducted here does illustrate an interesting alternative to some of the more established methods of applying CDs, such as the questionnaire approach (Kadoda 2000) and the activity profile approach (Britton and Kutar 2001). The characteristic of the approach we have taken here is that of abstract analysis, based on the physical properties of some notational medium. Furthermore, we give an example of analysing an actual electronic technology to determine how its features map onto the notational factors we have identified. It is the interaction between the technical constraints and the notational concerns that will finally determine the envelope of possible designs for our language.

Although this analysis process has been highly abstract, with no actual language envisaged at this stage of the research, we believe that the results will be of great value in guiding the kind of language we eventually develop, just as knowing the properties of engineering materials would be an important creative element in the design of novel engineering structures. The discussion in this paper has been unusually abstract, even for publications in the CDs field, but hopefully the reader will excuse the lack of concrete examples – to the degree that this promises an analytic alternative to current TUI design approaches that rely heavily on prototyping and evaluation to obtain similar results.

Acknowledgements

This research is supported by the Engineering and Physical Sciences Research Council grant GR/R87482 "Cognitive Ergonomics for Ubiquitous Computing" and by the European Union project IST-2001-34171 (WEBKIT) "Intuitive Physical Interface to the Web".

References

Aish R. (1979). 3D Input for CAAD Systems. Computer-Aided Design, 11(2), 66-70.

Bertin, J. (1981). Graphics and Graphic Information Processing. (Tr. W.J. Berg & P. Scott) Berlin: Walter de Gruyter.

Blackwell, A.F. and Hague, R. (2001). AutoHAN: An Architecture for Programming the Home. In Proceedings of the IEEE Symposia on Human-Centric Computing Languages and Environments, pp. 150-157.

Blackwell, A.F. (1996). Metacognitive Theories of Visual Programming: What do we think we are doing? In Proceedings IEEE Symposium on Visual Languages, pp. 240-246.

Blackwell, A.F. (2001). Pictorial representation and metaphor in visual language design. Journal of Visual Languages and Computing, 12(3), 223-252.

Blackwell, A.F. & Green, T.R.G. (2003). Notational Systems - the Cognitive Dimensions of Notations framework. To appear in John M. Carroll (Ed.) HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science. Morgan Kaufman.

Blackwell, A.F., Britton, C., Cox, A. Green, T.R.G., Gurr, C.A., Kadoda, G.F., Kutar, M., Loomes, M., Nehaniv, C.L., Petre, M., Roast, C., Roes, C., Wong, A. and Young, R.M. (2001). Cognitive Dimensions of Notations: Design tools for cognitive technology. In M. Beynon, C.L. Nehaniv, and K. Dautenhahn (Eds.) Cognitive Technology 2001 (LNAI 2117). Springer-Verlag, pp. 325-341.

Britton, C., and Kutar, M. (2001). Cognitive Dimensions Profiles: A Cautionary Tale. In G. Kadoda (Ed.) Proceedings of the Thirteenth Annual Meeting of the Psychology of Programming Interest Group.

Cockburn, A. & McKenzie, B. (2002). Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. Proc. CHI 2002, 203-210

Engelhardt, Y. (2002) The language of graphics. PhD thesis, University of Amsterdam.

Green, T. R. G. (1989). Cognitive dimensions of notations. In People and Computers V, A Sutcliffe and L Macaulay (Ed.) Cambridge University Press: Cambridge., pp. 443-460.

Green, T.R.G. & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' approach. Journal of Visual Languages and Computing, 7,131-174.

Green, T.R.G., Petre, M. & Bellamy, R.K.E. (1991). Comprehensibility of visual and textual programs: A test of superlativism against the 'match-mismatch' conjecture. In J. Koenemann-Belliveau, T.G. Moher & S.P. Robertson (Eds.): Empirical Studies of Programmers: Fourth Workshop Norwood, NJ: Ablex, pp. 121-146.

Ittelson, W.H. (1996). Visual perception of markings. Psychonomic Bulletin & Review, 3(2), 171-187.

Kadoda, G. (2000) A Cognitive Dimensions view of the differences between designers and users of theorem proving assistants. . In A.F. Blackwell & E. Bilotta (Eds.) Proceedings of the Twelth Annual Meeting of the Psychology of Programming Interest Group

Koike, H., Sato, Y., Kobayashi, Y., Tobita, H. & Kobayashi, M. (2000). Interactive textbook and itneractive venn diagram: Natural and intuitive interfaces on augmented desk system. Proc. CHI 2000, 121-128.

Lee, J., Su, V., Ren, S & Ishii, H. (2000). HandSCAPE: A vectorizing tape measure for on-site measuring applications. Proc. CHI 2000, 137-144.

Lertsithichai, S. & Seegmiller, M. (2001). CUBIK: A bi-directional tangible modeling interface. Proc. CHI 2001, 756-757.

Levinson, S. (1983). Pragmatics, Cambridge University Press.

MacEachren, A.M. (1995). How maps work: Representation, visualization and design: The Guildford Press.

Oberlander, J. (1996). Grice for graphics: pragmatic implicature in network diagrams. Information Design Journal , 8(6), 163-179.

Patten, J., Griffith, L. & Ishii, H. (2000). A tangible interface for controlling robotic toys. CHI 2000 Extended Abstracts, pp. 277-278.

Resnick, M., Martin, F., Sargent, R. & Silverman, B. (1996). Programmable bricks: Toys to think with. IBM Systems Journal, 35(3&4), 443-452.

Reuleaux, F. (1876) The Kinematics of Machinery: Outline of a Theory of Machines. Reprinted by Dover Publications in 1963

Suzuki, H. & Kato, H. (1995). Interaction-level support for collaborative learning: Algoblock – an open programming language. In Proceedings of Computer Supported Collaborative Learning '95. Lawrence Erlbaum.

Swift, J. (1726). Gulliver's Travels.