Cognitive Dimensions Tradeoffs in Tangible User Interface Design

Darren Edge and Alan Blackwell
Cambridge University Computer Laboratory

Draft Paper for CDs research workshop, Dallas Texas, September 24 2005

Introduction

As the tasks for which we use our computers become ever more complex and time-pressured, the need increases to find ways of interacting with computers that are more expressive, timely and direct than pointing-and-clicking. The conventional WIMP interaction paradigm – windows, icons, menus, and pointer – has a single point of focus (the pointer) operated by a single device (commonly a mouse), entailing only six fundamental interaction tasks: select, position, orient, path, quantify, and text [Foley et al. 1984]. In comparison, our everyday interactions with the physical world are many and varied, leading to a level of perceptual awareness and motor control far in excess of anything required by current generation interfaces. Utilising our spare capacity for interaction – the degrees-of-freedom and sensitivity of our hands, our bimanual skill, peripheral vision and spatial memory – requires a reinterpretation of Schneiderman's [1983] theory of *direct manipulation* that is more faithful than the 'indirect direct-manipulation' of WIMP interfaces. One such reinterpretation has given rise to the field of tangible interaction, in which the archetypical *tangible user-interface* of *TUI* (terms coined by Ishii & Ullmer [1997]) branches out into the 'real world' by representing informational and computational artefacts with physical tokens.

Although traditionally used to assist the design and evaluation of graphical user interfaces, Blackwell [2002] has applied the CDs framework to the design space of TUIs, accompanied by an analysis of the lexical, syntactic and semantic properties of tangible notational systems and the notational variables from which they are composed. This linguistic analysis of tangibles, drawing on previous work on graphical structures by Bertin [REF] MacEachren [REF] and Engelhardt [REF], led us to propose a new conceptualisation of TUIs as *manipulable solid diagrams* (REF rapid prototyping paper). This applies both to the individual physical tokens themselves as they are positioned in space and manipulated over time (assuming they have a spatial interpretation and mechanical degrees of freedom), and to aggregates of physical tokens as they are arranged in space and rearranged over time (assuming they have a relational interpretation and are spatially reconfigurable). This view is supported by the work of Stenning and Inder [1995], who say that "at some useful level of abstraction, different representations that are visually perceived in two dimensions can be regarded as the same medium". The perceptual basis for diagrams is the same, regardless of whether they are 'drawn' on paper, computer screens, or physical space. So whilst the physical nature of tangibles makes them *perceived* through a medium that has both visual and tactile qualities, they are still predominantly *interpreted* as a visual medium through the diagrammatic modality. The experiential qualities of use of a tangible token are conveyed mainly through the tactile medium that is the token itself, and these are interpreted through the tactile modality, but this aspect of tangible interaction is secondary to their use as a directly manipulable diagrammatic notation.

Whilst this metaphor is useful for analysing TUIs in terms of *how* they work, it is insufficient for TUI designers, who need to rationalise a priori *why* TUIs should work in a particular way. To do this requires an understanding of the many interacting factors that affect decisions in the design space of TUIs. Moreover, this design space is necessarily more complex than that of static solid diagrams and interactive graphical interfaces combined, due to the multitude of ways in which the 'tangible' layer of physical tokens can be coupled to the 'virtual' layer of informational and computational artefacts. It is therefore desirable to create analytic tools that TUI designers can use to identify the salient factors and trade-offs affecting choices made in this high-dimensional design space.

In this paper, we will explore further the application of CDs to the tangible domain, highlight distinctive trade-offs associated with certain classes of design decision, and relate this theory to existing TUIs, demonstrating why some aspects are as they are, and indicting some aspects that should be changed (PERHAPS).

The Notational Layers of Tangible User Interfaces

As stated in the introduction, TUIs can be seen as comprising of two high-level notational layers: the tangible layer and the virtual layer. The physical layer is a transient notation, in that the users do not create permanent marks in the same way as e.g. pencil marks on paper. Rather, the 'marks' of the notation are the spatial configurations of the tangibles, and more abstractly, the manipulation events that change their configuration over time.

This physical layer is complemented by a 'virtual' layer, through which computationally mediated feedback is conveyed to the user. This layer might make use of screen-based displays (similar to conventional workstations), projectors (commonly used to create large interactive surfaces), or solid state technologies such as LEDs and LCD displays embedded in the tangible tokens. The output of the system need not be limited to the visual channel either – a tangible interface can also give auditory feedback and tactile feedback of texture, temperature, weight and electric current (using TENS – transcutaneous electric nerve stimulation).

All tangible user interfaces must, by definition, have both a physical and virtual component. Even in a system such as the Actuated Workbench [REF] where there is no 'display' in a conventional sense, the computer controlled actuations of the tangible tokens acts as the output of the virtual layer. This can be seen as the physical extreme of tangible user interfaces, with regular WIMP (windows, icons, menus, and pointer) interfaces as the virtual extreme. Although conventional workstations would not generally be considered to be instances of tangible interfaces, they are the reification of a simple trade-off between many specialised devices versus a single generic device (the mouse). One of the main differences between a 'strict' tangible interfaces and an augmented graphical interface with many input devices concerns the relationship between representation and control. In a strict tangible interface (according to [REF]), the tangible elements embody mechanisms for representation and control, whereas in augmented graphical interfaces these are separated: representation is the screen output, and control is the device input. In terms of CDs, this can be seen as an issue of *role-expressiveness* – strict tangible interfaces express roles mainly in the physical layer through the form of their physical elements, whereas augmented graphical interfaces express roles mainly in the virtual layer through 'icons' and menu labels.

Whilst the tangible layer of TUIs can be viewed as a manipulable solid diagram, and to some extent analysed independently of the virtual layer, isolating the virtual layer of a TUI for independent analysis is not possible because its notation depends on its physical parts. Similarly, although it may be useful from an analytic perspective to break down the physical layer into the notations of individual tokens and the notations of token aggregates, from a design perspective this is not useful since choices made in one layer are very likely to affect the other. In TUIs, therefore, design decisions need to be holistic, taking into consideration the associated trade-offs that cut across all layers of analysis. This paper will proceed by introducing the cognitive dimensions themselves as they apply to TUIs, before detailing some of the distinctive trade-offs that exist in TUI design.

The Dimensions as Applied to Tangible User Interfaces

Provisionality: degree of commitment to actions or marks

The notion of provisionality can be interpreted in a number of different ways in the context of TUIs. Intrinsically, they can be seen as having lowest possible provisionality in terms of commitment to tangible instantiation, since you either instantiate a virtual object with a tangible token or you don't, whilst at the same time having the highest possible provisionality in terms of commitment to tangible interaction, since tangibles maintain an uninterrupted physical presence in the environment and can be attended to at any time.

In terms of the provisionality of actions made on tokens during the course of an interaction, these cannot be 'undone' in the same sense that a program can restore its previous state – any actions on tangibles can (in the absence of actuators) only be undone by manual reconfiguration. Moreover, human spatial memory is not perfect and so the 'undone' configuration is unlikely to be identical to the state which obtained before the offending action.

Progressive Evaluation: work to date can be checked at any time

The mechanical and spatial configuration of tangible tokens gives an indication of work in progress in those TUIs that physically embody the state of the system, but not of the steps taken to get there. These can only be conveyed through the virtual layer, and this reliance on the virtual layer for progressive evaluation increases as the proportion of system state embodied by the tangibles decreases.

Consistency: similar semantics are expressed in similar syntactic forms

The linguistic definition of consistency given in the CDs framework maps well onto the linguistic nature of solid diagrams, and so consistency in TUIs can be seen as using a single tangible syntactic structure for each semantic construct (e.g. using line-up and clustering to represent order and association respectively). However, consistency is not necessarily a good thing. Users might like to have a choice of different syntactic representations for similar semantics, based on the interactional qualities of the tangible syntactic structure. For example, stacks and line-ups [Engelhardt REF] can both represent token order, but their interactional profiles differ (see later REF). This notion of user choice has previously been raised in the context of CDs as the suggested dimension of *permissiveness* – multiple ways of doing things [in Blackwell et al. 2001].

Abstraction: types and availability of abstraction mechanisms

Abstractions are redefinitions that change the underlying notation in some way. Our characterisation of TUIs as manipulable solid diagrams gives us useful insights into the ways in which abstraction can be accomplished in the physical domain. Previous work on applying semantic concepts to graphical and sentential systems [REF] suggests that diagrams are successful as a form of expression not because they can express so much, but because they can express so little. The constraints on two dimensional spatial structures mean that only a small number of distinct relations exist, and these require less inferential effort in their interpretation than other, more complex representational systems. This 'information enforcement' is why diagrams are so pervasive in their use as an information modality, and similarly why conventional diagrams cannot express abstractions above a certain level of complexity. The wealth of general purpose programming languages, and the dearth of general purpose visual languages, clearly demonstrates this.

However, this situation only exists at the systems level of diagrams. Whilst many visual structures can be seen as 'taken' – Lakoff [1987] contends that certain preconceptual structures arising from human vision are given meaning by neuro-physiological logic (categories in terms of container schemata, hierarchical structure in terms of part-whole and up-down schemata, relational structure in terms of link schemata, and so on) – at the level of tangible tokens, it is likely that there are few such preconceptual structures, if any. Strong constraints still exist at the token level – there are only five ways in which two physical bodies can act as a 'joint' in a manipulable token (known technically as kinematic pairs, these are the screw pair, turning pair, sliding pair, cylindrical pair and spherical pair) – yet these still allow a rich vocabulary of abstract physical expressions to which users can assign their own meaning.

Physical abstractions in TUIs are therefore most conveniently implemented at the level of individual tangible tokens, within a fixed interpretative frame at the systems level of object relations. Although these may be represented physically, they still need to be managed through an interactive virtual layer which provides an abstraction manager.

Secondary Notation: extra information in means other than formal syntax

In those TUIs where spatial position is not interpreted, or where only the relative positions of grouped tokens are interpreted, rather than the relative positions of groups, spatial position can act as a useful secondary notation. For example, Sanders and McCormick [1987] describe a variety of strategies for the spatial arrangement of devices in the workplace – including arrangement by importance, function, frequency-of-use, and sequence-of-use – which are useful and convey meaning without any computational interpretation.

Physical tokens with certain material properties can also be directly annotated, for example using a marker pen on a wipe-clean surface. In fact, such a secondary notation may be necessitated by a TUI in which the virtual layer does not extend to the tangibles themselves, and in which physical tokens of the same type need to be identified.

Hard Mental Operations: high demand on cognitive resources

TUIs have an advantage over graphical interfaces in that they allow information to be externalised in physical tokens. The persistence of physical tokens means that they can act as mnemonic cues to the represented information, allowing cognitive offloading from working memory in the short-term, and reducing prospective memory failure in the medium to long term. Deeply nested goal structures are a further manifestation of *hard mental operations*, and these can be isomorphically represented and manipulated in the physical world by the syntactic structure of token stacks.

Viscosity: resistance to change

The arrangement of physical tokens in space is generally easy to modify and so has low viscosity, although it is dependent on the particular syntactic structures being manipulated, and in what way (see later REF). However, whenever the tangible notation utilises sequences of events in time, the viscosity becomes very high because such temporal sequences cannot be modified. The solution is to introduce temporal abstractions and a suitable abstraction manager in the virtual layer, but this introduces other trade-offs (see later).

Diffuseness: verbosity of language

In TUIs, both the positions of physical tokens in space, and the actions made on them in time, may form part of the notation. The language of a tangible notation may therefore have both spatial and temporal aspects, hence the verbosity of language – *diffuseness* – should also be considered from these two orthogonal viewpoints.

In terms of *spatial diffuseness*, tangibles have a fixed scale and so the level of physical detail is also fixed. However, the level of virtual detail can be dynamically adjusted, making TUIs a good choice for viewing detail in context. The size of physical tokens then becomes not a cognitive issue but a motor one: very small tokens are harder to manipulate. Tangible notations also have the property that the field of view is not necessarily limited to a single screen, meaning the spatial diffuseness is not such a problem as it might be, but introducing another set of trade-offs at the same time (see later).

The *temporal diffuseness* of an action is simply how long it takes, and in the physical world this is determined by the time taken to move the hands to the target token(s), arrange them in the desired configuration, and perform the action or manipulation. Factors influencing such actions are the spatial extent of the interface and the fit between the tokens' qualities and the user's interaction capabilities – commonly known as the token's affordances [Gibson REF].

Role Expressiveness: the purpose of an entity is readily inferred

The dimension of role expressiveness is best considered from a semiotic perspective [Peirce REF], with the entity interpreted as standing for something else – its referent. The more readily the link between entity and referent is inferred, the more role expressive the entity is deemed to be.

Within a TUI, there are three broad ways in which its physical and virtual elements can be imbued with meaning. The first is 'iconic' correspondence, where the sign demonstrates the qualities of its referent through literal, logical, or metaphoric similarity. The second type of correspondence is 'indexical', where the sign demonstrates the influence of its referent by directing attention to a certain spatial-temporal region. The final type of relationship between sign and referent is 'symbolic' correspondence, where the sign is interpreted to be a reference to its referent by appealing to an arbitrary law, rule or convention.

Physical tokens have the advantage over virtual representations in terms of iconicity, since they have a rich correspondence to other physical objects. Indexical correspondence, on the other hand, is best expressed through virtual forms, since these can change dynamically in the way that physical objects alone cannot. Note that an actuated physical token would be considered a hybrid physical-virtual representation in this respect. Lastly, the extent of symbolic crossover between the physical and virtual domains is such that neither has an advantage in terms of symbolic representation.

Closeness of Mapping: closeness of representation to domain

Closeness of mapping is a difficult dimension to interpret in a generic sense without any direct reference to the domain in question. One possibility is to evaluate the style of mapping used in the TUI and compare it to the result it is describing in the problem domain. Ullmer & Ishii [REF] posit that there are three styles of mapping in TUIs: spatial, relational, and constructive. Clearly, if the domain has a spatial component – i.e. it concerns the placement of objects in the physical world – a spatial mapping would be most appropriate. If the domain is abstract, based on symbols, logic and diagrams, then a relational mapping would be most appropriate. Finally, if the domain concerns the construction of objects in the real or virtual world, then a constructive mapping, where the result being described is built from the physical elements of the interface, would be most appropriate.

This is a simplification, however, and one can imagine real-world scenarios where all three styles of mapping are required. For example, consider a command and control centre used to monitor and direct the movements of 'units' (soldiers, emergency response vehicles, etc). To allow units to be manipulated in groups, and rapidly reconfigured, they might be built so as to connect together (constructive mapping). The position of groups of units in space is important (spatial mapping), but cannot be directly interpreted because the constructive assembly represents a logical rather than physical structure. Each group therefore needs to be associated with one point on the surface indicating where they are, and another point representing where they're going (relational mappings). The current position of the units is assumed to be outside of the control of the system, and so should be conveyed as some kind of virtual marker drawn on the map surface; the desired destination is a controllable variable, and so should be represented by some physical marker. Clearly, each TUI needs to be evaluated in terms of closeness of mapping within its context of use.

Premature Commitment: constraints in the order of doing things

Given that in TUIs the main locus of interaction is with physical tokens representing virtual objects, the fact that objects can generally only be manipulated through their tokens is an example of premature commitment concerning which virtual objects get instantiated as tokens and which don't. Premature commitment is also an issue when the notation relies on sequences of events: such sequences have a natural temporal order which acts as a strict constraint. This kind of premature commitment can be reduced by introducing temporal abstractions in the virtual layer, but at the cost of introducing additional trade-offs. These will be discussed in the next section.

Visibility: ability to view components easily

Any notations that are transient, for example by making use of space and time as their medium, have inherent problems with visibility since the current state of the notation represents only a temporal 'slice' from a sequence that cannot be recovered unless it has been recorded in some way. One way of doing this is to retrieve the meaningful events from the sequence and translate their temporal order into a spatial order in the virtual layer, e.g. by laying out frames on a surface. This is the basis of another trade-off to be discussed in the next section.

Hidden Dependencies: important links between entities are not visible

Links between entities can be represented at both the token and the system level of diagrams. At the physical token level, morphological similarity can indicate fixed relationships, but this requires some prior assumptions concerning the number and type of such relationships. The dynamic nature of the virtual augmentation of physical tokens solves this problem. At the system level, the problem of physical links mirrors the problem of link representation at the physical token level: prior assumptions are required, and furthermore the physical links may be susceptible to entanglement (as in the system by Patten, Griffith and Ishii [2000]). There is no convenient physical equivalent of the graphical node-and-link formalism: spatial clustering can only indicate undirected n-ary association within token groups; line-up can only indicate directed linear associations; and there is no way to indicate specific properties of binary relationships. For the representation of trees and networks, therefore, the virtual layer should support the physical layer by tracking token positions and dynamically updating the virtual links drawn between them.

Error Proneness

TUIs have added potential for errors than graphical interfaces alone: physical tokens can accidentally be knocked out of position, placed outside of the sensed area etc., and there is no simple way to recover previous tangible state. Current methods of tracking token positions, configurations and identities are also far from perfect, and so to some extent TUIs designed to work in this technological environment need to be built with robustness in mind, providing facilities to recover gracefully from both technological and interactional errors.

CDs Trade-offs in TUI Design

This section will ground the above discussion of how CDs can be given a tangible interpretation by introducing the notion of *design decisions*, and showing how seemingly simple choices can have multi-dimensional trade-offs which cut across notational and architectural layers.

Starting at the highest level of analysis, we begin by discussing CDs trade-offs at the virtual level of TUIs.

Virtual Trade-offs

Abstraction of Time

The abstraction of time has been a recurring theme in many of the dimensions discussed previously, and is one of the most significant design decisions. Assuming that the notation at least some temporal features (otherwise it would be very primitive), the possibilities range from having no virtual representation of event sequences, to having a notation where time is represented in the virtual layer, either temporally as video sequences or spatially as event frames.

Having no representation of time avoids the problem of *premature commitment* that affects systems relying on temporal abstractions, which require action sequences to be planned completely before they are executed. However, when there is no temporal abstraction there is no way to refer to what has been done – a problem with *progressive evaluation*. Consequently, previous work cannot be undone – a problem with *viscosity* which also leads to *error proneness*. Such systems are only be useful for simple systems with no need for temporal abstractions. At present, this represents the majority of TUIs presented in research papers (Evidence?), which suggests that these constraints are not as severe as a CDs analysis might suggest.

Using videos (or macros) in the virtual layer to represent interaction sequences has the advantage of allowing sequences to be stored and reused – a useful *abstraction* facility. This approach also has the virtue of *consistency* between input and output, but the consequences are mainly negative. First of all, there is the problem of *visibility* – only one frame of each video can be seen at a time, and if multiple videos are to be compared, this either results in *hard mental operations* by forcing the user to remember the contents of the entire first video when viewing them sequentially, or in a *juxtaposability* problem (side-by-side visibility) in that watching two multiple videos in parallel requires divided attention. There is also a problem of *viscosity* – although such sequences can be stored and reused they are not easily modified.

Using event frames in the virtual layer has a number of advantages over the other two approaches, including better *progressive evaluation* (previous work is *visible*), reduced *viscosity* (event sequences can be modified spatially), and increased *juxtaposability* (it is easier to compare frames in space than in time). Where temporal *abstractions* are required, this may well be the best approach provided the interaction sequences can be split into events and represented conveniently in the virtual layer.

This discussion of time raises the issue of whether the tangibles themselves are the information structure being created, or whether they are simply being used to construct some virtual information structure such as a temporal abstraction. This issue, of *embodiment* concerns the coupling between the virtual and physical layers, and will be discussed next.

Virtual—Physical Trade-offs

Degree of Embodiment

If the virtual output of a tangible interface is overlaid on its physical tokens (e.g. by a video projector), then the two can be viewed simultaneously and manipulated as one. The interaction is embodied, in that the focus of the interaction is on the tokens being attended to [Dourish REF]. This superimposition is an interesting variant on the regular cognitive dimension of *juxtaposability*, which considers the ability to view components side-by-side. However, superimposition may itself lead to a *visibility* problem, in that the physical tokens or arms of the user may obscure part of this visual overlay. Moreover, widely distributed tokens means widely distributed attention when visual output is on or around the tokens themselves.

A single region of visual interest – distinct from the tangible elements – gives greater visibility to the information that requires constant visual attention. However, this then leads to reduced *juxtaposability* between physical and virtual elements, and hence reduced *embodiment*.

A recommended design strategy in this instance is to focus attention on a visual display of the most important information, whilst monitoring the physical tokens (and potential virtual overlays) using peripheral awareness. This strategy corresponds to the Weiser's vision of calm technology, where he states that [Weiser & Brown 1996]:

> "Technologies encalm as they empower our periphery. This happens in two ways. First … a calming technology may be one that easily moves from center to periphery and back. Second, a technology may enhance our peripheral reach by bringing more details into the periphery"

Degree of Synchronization

When a tangible interface continuously tracks and interprets the configurations of its elements in real time, it allows users to *progressively evaluate* their actions, but these must be made in strict sequence resulting in *premature commitment.* If, however, the elements of the interface are sensed and interpreted in batch mode, it allows users to experiment *provisionally* with configurations of elements before committing to their 'epistemic' actions – those actions performed to uncover information that is hidden or hard to compute mentally. Fitzmaurice [1996] states that such epistemic actions allow users to externalise information in a manner that reduces the number and/or complexity of *hard mental operations* in two distinct ways: by reducing the memory involved in mental computation (space complexity), and by reducing the number of steps in mental computation (time complexity). Furthermore, epistemic actions can also reduce the probability of error of mental computation (unreliability), resulting in less *error proneness.* However, by not giving continuous feedback to the users, errors may go undetected for longer. Therefore epistemic actions can be seen to trade off true errors resulting from incorrect mental computation, against *slips* (as they are referred to in Cognitive Psychology) resulting from executing the correct actions in an incorrect way (and not realising).

A recommended design strategy in this instance is to continually track and interpret tangible elements in real time, but introduce an additional notational layer in which users can indicate their degree of commitment to their actions or marks.

Degree of Coherence

Not all physical objects in TUIs are token-like, i.e. permanently mapped to the same virtual object. A continuum exists, which Koleva et al. [REF] call the *degree of coherence.* In the case of fixed coupling between physical 'devices' and virtual 'variables', there is a no *abstraction barrier* since the interface is approaching true direct manipulation. As soon as the coupling becomes in any way transient, an *abstraction facility* is required in the virtual layer with which to make assignments between the devices and variables. As variables are 'instantiated' as devices rather than appearing on some display as icons and menu items, there can be increased *visibility* of content relative to controls. In turn, this may require a *secondary notation* to distinguish between similar input devices (secondary notation is discussed by Petre & Green [1992] and Oberlander [1996]).

For a TUI comprising a fixed set of variables, the closer it is to the space-multiplexing extreme (one-to-one correspondence between device and variable), the greater potential there is for each device to be more *role-expressive* than would be possible with fewer devices. This is at the cost of greater *diffuseness*, because the number of devices increases with the number of variables. The closer such an interface is to the time-multiplexing extreme (one-to-many correspondence between device and variable), the less diffuse the interface because there are less physical devices. This is at the cost of a weaker mapping between device and variable, since a single device is unlikely to map to different variables equally well.

A recommended design strategy in this instance is to have a suite of strong specific devices for common or complex variables, and complement these with weak general devices for the remaining variables.

Token Aggregation Trade-offs

Representation of Token Association

In a tangible interface of multiple objects (physical objects representing virtual ones), there are two broad ways in which association and disassociation can be represented. The first is object-to-object relations, such as spatial clustering (associated objects are close to each other), separation by a separator (disassociated objects are separated from one another), and containment by a container (associated objects in the same container). The second is attribute-based relations, where associated objects have a similar size, shape, colour, etc.

With object-to-object relations, there is a marked *juxtaposability* between disassociated objects. This characteristic is only weakly present in attribute-based relations, since disassociated elements can still have close spatial proximity. Similarly, object-to-object relations make their dependencies explicit in a way *consistent* with common graphical syntax (e.g. a container owns its objects, a separator defines two classes of object), whereas attribute-based relations can be arbitrary and suffer from *hidden dependencies* (e.g. multiple shapes of multiple colours has no obvious interpretation but could encode multiple relationships).

An advantage of attribute-based relations is that they can represent *abstract* relationships that are difficult to express using object-to-object relations. For instance, coloured illumination can be used to express the set inclusion relationships between multiple objects and multiple sets – something only possible in object-to-object relations by constructing some tangible form of Venn diagram.

A recommended design manoeuvre in this instance is to use whichever representation provides the better *closeness of mapping* for the particular problem domain. For example, if an ownership relation in the domain is strong (i.e. composition), then an object-to-object relation should be used (e.g. containment), whereas if it is weak (i.e. aggregation), then an attribute-based relation should be used.

Representation of object order

In a tangible interface of multiple objects, there are two broad ways in which object order can be represented. The first is through line-up, where the objects are ordered along the x- or y-axis on a planar surface. The second is through stacking, where the objects are ordered on top of one another to form a column rising along the z-axis.

From the perspective of *diffuseness*, stacking is superior to line-up since it takes advantage of the limited third dimension (hence tangible interfaces are often called 2.5D), freeing up planar space for alternative use. Stacking also makes certain *dependencies* explicit. For example, if the bottom object is moved on the planar surface, then the objects resting on top of it will move accordingly. Similarly, if an object is removed from the middle of a stack, the higher objects will move, under gravity, to fill the gap. These behaviours do not occur with line-up: the dependencies are hidden.

Advantages of line-up over stacking include *visibility* and *error-proneness* – stacked objects are difficult to see and easy to knock over. *Viscosity* is also an issue, but manifests itself in two distinct ways. Stacks have low *movement viscosity* – their nature means that all stacked objects can be moved as one – but high *manipulation viscosity*, since they require great dexterity or many sequential operations to modify their order. This is the inverse of object line-ups, whose order is easy to change through simultaneous bimanual actions in the plane (they have low manipulation viscosity), but which cannot be moved so simply as a unit without breaking the structure (they have high movement viscodity)

With this design decision, as with the others, it is necessary to make a design manoeuvre along the set of fundamental trade-offs, or adopt a design strategy that in some way overcomes them (at the cost of introducing other trade-offs). This should be done based on the characteristics of the environment (e.g. space, number of tangibles) and the profile of notational activities to be performed (e.g. degree of search, modification etc.).

Token Trade-offs

Representation of Continuous Values

With a tangible interface that operates as an interactive surface, the most salient dimensions of information are likely to be assigned to the 2.5 spatial dimensions due to their *visibility* and potential *closeness of mapping*. The next most obvious way to encode continuous information is through the orientation of objects on the surface. Yet, most interesting systems have more than 3.5 dimensions of information. So we need to find additional ways of tangibly representing continuous values.

One way of approaching this problem is to look at kinematic pairs of elements [Reuleaux 1876] as potential carriers of information. These are defined as two bodies that reciprocally envelope the relative motion of the other, leaving 1, 2 or 3 remaining degrees of freedom. The prototypical 1D example is a screw pair, consisting of a nut and bolt, whose relative movements describe a helical path. By mentally simulating the effect of allowing the pitch of the screw thread to tend to zero degrees (relative to the nut), a revolute pair is formed which allows only rotational motion. Similarly, by allowing the pitch to tend to ninety degrees relative to the nut, a prismatic pair is formed which allows only translational motion. Hence, screw pairs, revolute pairs and prismatic pairs are the fundamental constructions for physically representing 1D values in tangible interfaces, and twisting, turning and sliding are the fundamental actions for manipulating them. Similarly, 2D values can be represented by a cylindrical pair, which allows a cylinder to rotate within and slide along a cylindrical cavity, and 3D values can be represented by a spherical pair, which allows a ball to rotate in all three directions within a socket.

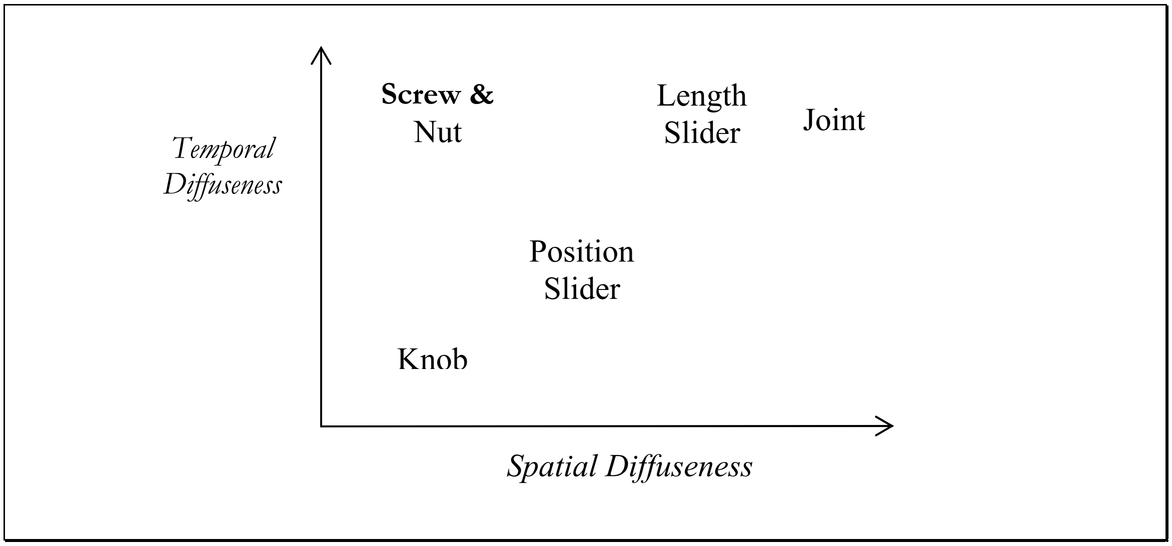
In order to make a cognitive dimensions comparison of the three 1D kinematic pairs as information artefacts, it is necessary to make a distinction between *spatial diffuseness of expression* and *temporal diffuseness of action*. It is also necessary to distinguish between simple expressions of each of the three 1D kinematic pairs. A sliding pair can take the form of one prism sliding within the confines of the other – a 'position-slider' – or of two similar prisms sliding relative to one another, making a 'length-slider' (or 'telescopic' tangible, as in the 'stretchable square' of Fitzmaurice, Ishii & Buxton [1995]). A turning pair can be a direct rotation device – as in a 'knob' – or an indirect rotation device, where rotation of the 'joint' is a consequence of the movement of the joined elements (an 'articulated' tangible). For twisting pairs, either the 'screw' or the 'nut' can be fixed in position, with interaction taking place on the other, moveable element. Other simple expressions of the basic kinematic pairs are possible, but I will focus on these for the purpose of illustration.

A position-slider, due to its linear and compact form, can operate side-by-side with identical devices, all of which can be operated near-simultaneously with simple one or two-handed movements. A length-slider is clearly more spatially diffuse, due to its varying size, but also takes more time to operate due to the interaction requiring two hands.

A knob (utilising a single revolution to convey value) is even more spatially compact than a slider, and takes a similar amount of time to operate. A joint requires two hands to operate and so has a similar temporal diffuseness to a length-slider, but takes up varying amount of area depending on the joint angle and so is relatively more spatially diffuse.

Twisting devices – both screws and nuts – are less spatially diffuse than sliders due to the use of an extra dimension when expressing the single degree of freedom (think of a twisting pair as a coiled-up position-slider). However, they also take longer to operate (in general) than the other two types of kinematic pair, due to the many rotations required to achieve a given translation.

The situation can be visualised as shown in Figure XXX:



~ Figure XXX: Comparison of the diffuseness of 1D kinematic pairs ~

There are also non-diffuseness trade-offs associated with these kinematic pairs. Joints are the only ones that can easily be composed into a linkage, which may provide a facility for higher-level *abstractions*. Length-sliders are the most *role-expressive*, as they represent quantity by their physical size. Twisting pairs (screw & nut) have the property of being difficult to change quickly, making them less *error-prone* than other forms of manipulation. Position-sliders have a greater degree of *juxtaposability* than the others do due to their linear form affording side-by-side comparison, but suffer from *premature commitment* due to their bounded range.

In the above analysis with a single revolution of a knob being used to convey value, it was shown to have the lowest temporal and spatial diffuseness of the kinematic pairs considered. However, the special quality of a knob is that it can be dynamically tailored to operate in different ways. A simple knob (without any feedback) can only convey values in a bounded range equivalent to a single revolution – essentially equivalent to a position-slider. A knob can be augmented by a radial array of lights (as is done on some electronic midi devices) to track the number of revolutions completed, essentially simulating the rotation and linear progression characteristics of a twisting pair. A more *abstract* additional notational layer, capable of conveying any number of revolutions, allows the knob to fully exploit its free-turning property by allowing an unbounded range to be represented. However, by not committing to any particular range, getting to any value in particular can be a time-consuming or impossible affair if the angular increment of the knob is not set at the correct level.

A recommended design strategy in this instance is to use multiple devices to control the value of a single variable. One possible scheme uses a 'gear' metaphor of two controls: one to increment the value, and another to change the size of the increment. This approach was used in the SeismoSpin device [McKelvin et al. 2003] to navigate time on a scale of minutes to decades. Another, 'hierarchical' scheme utilizes multiple controls to control varying increments of the coupled variable. Note that whilst such schemes may result in faster and more accurate control, the physical devices themselves no longer embody the digital state of their variable, introducing another trade-off between *temporal diffuseness* and *abstraction*.

To consider:

Number of token types
Representation of object properties
Coupling of representation to control
Structural approach: Interactive Surface, Constructive Assembly, Token+Constraint:
TAC: Tokens need to fit constraint
Once committed to TAC, can't elaborate tokens with structural features etc.
With tangibles, abstraction potential is decreased by iconicity, whilst role-expressiveness is increased.
Restoration of stored state

To Do:

References
Ensure consistent terminology is used
Examples and case-studies
General design recommendations
Conclusions