Extending Direct Combination: recursion, abstraction and other extensions

Simon Holland

Computing Department, The Open University, Walton Hall,
Milton Keynes, MK7 6AA, United Kingdom
S.Holland@open.ac.uk

Abstract
Direct Combination is a user interaction framework that can be applied to desktop computing and to mobile and ubiquitous computing to lessen the user's need to search the user interface in certain circumstances. This paper investigates various ways of extending Direct Combination, including recursion, composition, abstract objects, and ambient combination afforded by hand gestures. This paper assumes that the reader is already familiar with the basics of Direct Combination.

1 Introduction: extensions to DC

DC is a user interaction framework designed to afford users the freedom to apply simple interaction styles (such as pairwise interaction) to reduce the amount of search and navigation required to operate user interfaces. DC appears to be useful when search is expensive for the user- e.g. when the user is distracted, or the task or objects involved are unfamiliar, or the user interface or environment is complex. Part of the attraction of DC is the simple, minimal nature it can lend to user interaction in complex situations. This paper discusses various extensions to Direct Combination. Some of these extensions make some interactions more complex than simple DC. Some of the extensions may not have the same capacity to reduce search as simple DC. Simple Direct Combination is applicable in a wide variety of contexts - for example, desktop computing, tangible computing, ubiquitous computing and mobile computing :some of the proposed extensions are not so widely applicable. However, the techniques all cast useful light on practical and theoretical aspects of DC. The techniques are as follows.

- Recursive DC
- DC Composition
- Abstract Objects
- Wandless, gestural AC
- Universal gestures

This paper assumes that the reader is already familiar with Direct Combination (Holland and Oppenheim, 1999) and Ambient Combination (Holland , Morse and Gedenryd, 2002).

2 Recursive DC

A useful exploratory heuristic with any new computing technique is to consider whether it can be applied recursively. Recursive DC may be introduced by means of an example. As a (different) heuristic technique for exploring the capabilities and limits of DC, an active search was carried out for domains that appeared to be *unsuitable* for the application of DC. One domain considered as part of this process was Word Processing

(WP). Word Processing appears to be an unpromising domain for applying DC for the following reasons.

1) At first sight, there is no great variety of objects of interest visible in WP. The most visible objects of interest in WP tend to be 'pieces of text'. In general, a variety of objects of interest typically need to be present in a domain for DC to be useful[1].

2) The most obvious visible objects of interest in WP - for the sake of argument let call these 'selections of text' - tend to have many attributes of interest- e.g. font, font size, emphasis, style, string content, ruler settings, justification, outline level, ect. In general, when objects of interest have a large number of attributes of interest, and these objects are combined for DC purposes, there is a danger that this may create a combinatorial explosion of possible operations. For example, consider a case where the user wishes to apply the font style from one text selection to another. If pairwise interaction is applied to the two selected pieces of text, there is a danger that the relatively large number of attributes of each object may give rise to a relatively large space of possible operations, defeating the purpose of reducing the search space.[2] The notion of recursive DC hinges on the fact that by adopting a very slight change in point of view, the problem of 'too many attributes' can easily be turned into a positive new source of affordances, as follows. Recall firstly that the principle of DC demands that all objects of interest should be visible and capable of manipulation, and that any object of interest should in principle be capable of interaction with any other object of interest. *All that is required is:*

- *to recognise that interesting attributes that belong to objects of interest should themselves be recognised as objects of interest.*
- *to provide suitable affordances for manipulating the attributes as objects.*

So for example, consider the case of two selected pieces of text, where the user wishes to use the font from one piece of text for the other. All that is required is for the user to be able to open a window on the object (here the selected text) and look at its labelled attributes or sub-parts, (e.g. font, font size, emphasis, style, etc) using a tool something like a Smalltalk or Self object browser. If the user's aim is to apply the font style from one text selection to another, then all that is needed is to drop the font style object from inside the first text object onto the second (unopened) text object. The principle that *interesting subparts of objects of interest should themselves be treated as objects of interest* is perhaps obvious, but it is powerful and generally applicable (like most applications of recursion). Consequently it is useful to state this principle explicitly, even though it might be considered to be inherent in the existing principles of Direct Combination. In any case, the uniform application of this principle is not automatic, since it requires the provision of interaction techniques to allow users conveniently to gain access to subparts of objects. This is relatively straightforward in the case of desktop DC (ref) , tangible DC (2002) , and ubiquitous computing (reference) - since in none of these cases is feedback bandwidth generally particularly scarce. However, in the case of mobile computing, such internal access makes demands on display space either on the

[1] From another point of view, there are diverse objects of interest in word processing: styles, fonts, outline levels, rulers, headers, footers - but this does not affect our use of WP to illustrate recursive DC.

[2] A different reason why WP is not obviously an attractive domain for DC is that typical users of WP tend to become skilled and familiar with existing ways of doing things - minimizing the incentive to learn new interaction techniques - but this does not affect our argument.

selected object in the environment, or on the user's feedback device - which is typically scarce. One small technical aside is worth noting. If a user wished to apply the font *size* from one text selection to another, transferring *a raw integer* representing the font size would be ambiguous, and consequently not a good idea. Recursive DC is best supported where good programming practices are followed - i.e. font size would be represented not by a raw integer, but by a *font size* object.

3 DC Composition

When any new computing technique is developed, parsimony demands that one consider whether the technique can be composed, i.e. whether the product of one application of the technique can be used as the input for another application of the technique.
In the case of DC, the simple-minded literal answer to this question is no. This is because the input to a DC interaction is the set of selected objects, and the output is an ordered set of possible actions, with a default selected. Thus, given that the inputs and outputs of a DC interaction are not of the same kind, it is generally not possible to compose DC interactions. However, if we allow some leeway in interpretation, DC interactions can be composed in some cases in the following sense. When using DC, after a user has selected some objects and has been offered a set of options, a user may respond in any of the following ways:

- execute one of the offered commands,
- refine or amend a command and then execute it,
- cancel one or more selections,
- take no action.

In some cases, the execution of a command may result in the creation of a new virtual object, represented on some relevant feedback device, or a significant change in the state of some existing object of interest In such cases, the user might select this object as the subject of a further DC interaction. In this sense of the word, it can be extremely useful to compose DC interactions. Examples where this might be useful might include creating collections of objects - see the section on "Abstract Objects". Because the creation of a new virtual object makes demands on the relevant feedback devices, this particular technique appears to be more readily applicable to desktop DC and ubiquitous computing, where feedback bandwith is relatively plentiful, rather than mobile computing, where it tends to be scarce.

4 Abstract Objects

On the desktop, and in ubiquitous environments with lots of display space, it may often be to the advantage of users for objects of a relatively abstract nature to be represented concretely (for example, file directories or folders). One place to display objects is on mobile device displays, although display space there can be scarce. Consequently, even though abstract objects do not generally naturally occur in a scannable form the environment, we have been exploring the representation of various objects via such means as wallcons (scannable labelled icons with id markers, arranged on a wall to repesent diverse frequently used resources) and resource murals (collections of wallcons). Such displays could be projected and programmable, or static and physical. We have established that the former can be scannable in some circumstances. Such use of display space in the environment is not to everyone's taste, although the large amounts of unused

display real estate present of the walls of many offices, meeting rooms and places of work are tempting to some. Leaving aside the techniques used to display the objects, but noting that such techniques exist, let us consider examples of the value of abstract objects in Direct Combination. Since our current prototype DC systems tend to be limited to pairwise interaction rather than full n-dim interaction, collections are one good example of useful abstract objects.Two other candidates for abstract objects are the 'Collection of relevant documents' wallcon for meeting rooms and the 'On' and 'Off' wallcons.

Scenario1 (collection of documents, this pda)
This scenario involves the "Collection" abstract object. The aim of the scenario is to illustrate the principles involved, rather than to champion details of how the various parts of the interaction should be afforded. Note that the PDA featured is assumed to provide in general only pairwise DC, not n-fold DC. Anne is at a poster session at a conference and sees several posters which interest her amongst a much larger collection on the same wall. Anne would like to email copies of, or retrieve information about, these selected posters to herself. Anne would like to minimize the amount of time and effort required to achieve this end. Anne selects the Collection object on a strip on her PDA and then zaps the interesting posters in turn. Because Anne has selected the collection object first, the rest of this interaction is treated specially as an n-fold, rather than a pairwise combination. The default offered operation is 'Add zapped objects to new collection', in response to which Anne presses 'Do it'. By convention for this operation, the resulting collection is already selected ready for the next combination (it can be cancelled if not subsequently wanted). However, in addition,the newly created collection of documents is displayed in the temporary area on her PDA as a temporary resulting object, in case, for example, Anne wants to alter its contents or use it later. To select the second object for her next combination, Anne then presses the' include this PDA' button (one of whose standard roles is to represent Anne, its owner). The options offered are

Email Collection of documents to self,
Email selected documents from collection to self,
Email Collection of URLS associated with collection of documents to self.

Scenario 2 (collection of documents, printer)
Anne attends a curriculum group meeting and find that she does not have a copy of the agenda or the discussion document. Anne zaps the 'Relevant documents' icon on the resource mural strip on the wall of the meeting room, and zaps the printer in the corner. Anne is presented with the following choices.

Print collection of documents for M206 meeting to meeting room printer
Choose documents from collection to print to meeting room printer.

Scenario 3 (Sundry devices, "Off" wallcon)
Anne needs absolute quiet in the meeting room for her demo but there seem to be a lot of noise machines in the meeting room. Anne goes around with her wand pointing at the coffee machine, and the printer and the fan and various other devices switching them off via the menus presented on her wand for each device. One obscure machine to do with the air conditioning presents problems. Selecting the machine causes the wand to offer

screens of technical sounding commands, but no obvious off command. Ane zaps the obscure machine, then zaps the abstract "off" icon on the wall (next to the "on" icon) and presses "do it". The machine switches off. The display of large general purpose 'on' & 'off' wallcons in work places, though strictly redundant, could be useful for safety reasons - for example in case the feedback screen on a device is broken, for use in emergencies, and for the convenience of those with difficulty reading small screens. Alternatively, Anne could have held down the collection button, zapped all of the devices in turn and then zapped the off wallcon to switch them all off at once.

Interestingly, isolated examples can be found in everyday life of interactions very similar to the scenarios using collections illustrated above. For example when buying items at a supermarket till, the laser scanner may be viewed as implicitly forming an n-fold collection of items to be purchased. When a credit or debit card is swiped in the magnetic scanner, the card effectively forms a pairwise combination with the collection of purchased items, and a single action is offered - purchase the goods. Some supermarkets allow another option: "cash back", where in addition to purchasing the goods, the customer may withdraw money. The interaction is driven almost entirely by zapping items. Of course, the supermarket interaction is rigid and narrow, whereas Ambient Combination provides a broad and flexible framework for a rich variety of such interactions

5 Universal gestures

One key insight in the design of the Xerox Star, which became a cornerstone of desktop GUI interaction design, was the use of 'universal commands': in other words, relatively abstract verbs such as print, copy and delete that could be applied polymorphically to diverse objects. Because the same small number of verbs could be applied in so many different contexts, the user's cognitive load and burden of memorisation could be greatly reduced, and the range of quickly and easily affordable interactions increased. By contrast with desktop GUI interaction, there is no consensus yet on the key elements of future frameworks for mobile and ubiquitous interaction design. Many researchers have investigated ways in which hand gestures (or other bodily gestures) might play various roles in mobile and ubiquitous interaction. Assuming that appropriate gesture capture technologies will become convenient and economical to use, we propose

- the creation of universal gestures that can be applied in a wide variety of contexts in ubiquitous computing (this idea is unlikely to be original to us)
- ways in which universal gestures could work usefully with Direct Combination.

Gestures in a ubiquitous environments are particularly expressive since they can be used not just to express verbs (universal commands) but also to express nouns, by pointing at objects in the environment.The next section demonstrates how such a capability could be used within the AC framework for a rich and expressive range of interactions. Below we give a sample of some possible universal gestures for use in ubiquious environments, mapping gestures to meanings (table 1). This sample is illustrative - no special claims are made for the appropriateness of the specific choice of universal commands, nor the gestures assigned to represent them.[3] Various technologies have been predicted that could

[3] Some other candidate commands include open/look inside, yes/no, on/off and others.

make this form of interaction possible: wearable position sensors, sensing based on natural biological electrical fields and room-based sensing, for example by video. Sensing technologies could be coupled with interpretation technologies such as neural nets. As already noted, gesturing becomes particularly expressive when it is possible to use it to identify objects in the environment by pointing. This could be possible with an architecture that combined indoor location and orientation sensing with gestural recognition, or in some cases,video technology.

Command	**Gesture**	**Alternative gesture**
select	point and clap	point and click fingers
cancel/stop	horizontal chop	cut throat gesture
do it/go /start	rotate finger	divers OK gesture
increase	palm up, arm angled up	
decrease	palm down, arm angled down	
next	palm vertical wave right	
previous	palm vertical wave left	

Table 1

6 Ambient Combination by hand gestures

By putting the AC framework together with gesture recognition and universal gestures, it is possible to create wandless, gestural AC. For example consider a meeting room with the ability to recognise its occupants (e.g. by active badges) and to recognise universal gestures made by its occupants (e.g. by video processing), and which has a display on each wall for use as a feedback device for public AC interactions. Note that because of subsumption, commands may be specified in a variety of ways, e.g. via pairwise interaction, or by selecting a single object then using a universal gesture.

Scenario1 (self, other person)

Anne arrives early for the meeting. Only one other participant, a stranger has already arrived. While waiting for the other participants to arrive, they decide to exchange cards. Anne selects herself and the stranger using universal *selection gestures*. Since they are strangers, the option offered is

Anne Dante send card to Joe SixPack

As this is the only option offered on her private feedback display , Anne simply gives the universal "do it" hand gesture.

Scenario2 (self,document)

Looking around the room. Anne finds one of the paper documents on display relevant to her interests. Anne indicates the document and herself using universal *selection gesture*. Her provate feedback device offers the following options

Email brochure to anne dante

Email URL of brochure to anne dante

Anne uses the "next" hand gesture to choose the second option then gives it the the "do it" gesture"

Scenario3 (documents for meeting, printer)

Anne realises that she does not have the briefing documents for the meeting. Anne selects the rooms "Documents for current meeting" icon on the wallscreen with a hand gesture and selects the printer in the corner.
Anne's feedback device offers the following options

Print collection of documents for M206 meeting to meeting room printer
Choose documents from collection to print to meeting room printer

Anne uses the "next" hand gesture to the feedback screen to choose the second option then uses the next and OK gestures to pick documents to be printed.

Scenario 4 (*lights, projector, curtains: unary actions)*
Much later, having finished her presentation, Anne makes the select hand gesture to the data projector and gives it the "off" hand gesture to switch it off.
Anne makes the select hand gesture to the lights and makes the on hand gesture to switch them on. She makes the select gesture to the curtains and uses the 'do it' gesture to make the curtains to open themselves (the first option available).

7 AC and voice recognition
In some situations where voice recognition is available, it may still be more convenient or more reliable to select objects in the environment by pointing at them rather than naming them. Under AC, given the principle of subsumption, when a user employs n-fold interaction to help specify a command, but has not yet chosen a verb, the user may, amongst other possibilities, specify the verb explicitly. Hence, one interesting special case of AC would involve selecting one or more objects in the environment by pointing and specifying the verb by speaking. This would be an alternative to using a device such as a stylus or keyboard to explicitly type in a verb, or to searching the options displayed on a feedback screen for the desired verb . Of course, given the principle of subsumption, it would still be possible to issue complete commands using voice recognition where this was convenient.

References
Holland, S. (2002) Ambient Combination: The application of Direct Combination to Mobile and Ubiquitous Human Computer Interaction. Technical Report TR2002/1, Department of Computing, The Open University, Milton Keynes MK76AA Great Britain.

Holland, S. and Oppenheim, D. (1999) Direct Combination. In *Proceedings of the ACM Conference on Human Factors and Computing Systems CHI 99*, Editors: Marion Williams, Mark Altom, Kate Ehrlich, William Newman, pp262-269. ACM Press/Addison Wesley, New York, ISBN: 0201485591.