Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI

Simon Holland, David R. Morse & Henrik Gedenryd

Computing Department, The Open University, Walton Hall,
Milton Keynes, MK7 6AA, United Kingdom
Fax +44 1908 652140
{S.Holland, D.R.Morse, H.Gedenryd}@open.ac.uk

Abstract. Direct Combination (DC) is a recently introduced user interaction principle. The principle (previously applied to desktop computing) can greatly reduce the degree of search, time, and attention required to operate user interfaces. We argue that Direct Combination applies particularly aptly to mobile computing devices, given appropriate interaction techniques, examples of which are presented here. The reduction in search afforded to users can be applied to address several issues in mobile and ubiquitous user interaction including: limited feedback bandwidth; minimal attention situations; and the need for ad-hoc spontaneous interoperation and dynamic reconfiguration of multiple devices. When Direct Combination is extended and adapted to fit the demands of mobile and ubiquitous HCI, we refer to it as *Ambient Combination (AC)*. Direct Combination allows the user to exploit objects in the environment to narrow down the range of interactions that need be considered (by system and user). When the DC technique of pairwise or n-fold combination is applicable, it can greatly lessen the demands on users for memorisation and interface navigation. Direct Combination also appears to offers a new way of applying context-aware information. In this paper, we present Direct Combination as applied ambiently through a series of interaction scenarios, using an implemented prototype system.

1 Introduction: Problems with Mobile HCI

User interfaces for mobile devices must typically deal with four general problems. Firstly, only limited screen real estate is available; more generally, taking into account non-visual forms of feedback such as auditory displays, *feedback bandwidth* is limited. Secondly, the bandwidth, precision and convenience of *input devices* are generally restricted. A third problem is that many mobile devices are typically used in *minimal attention situations* [1], where the user has only limited, intermittent attention available for the interface. In such situations, interactions with the real world are generally more important than interactions with the computer, the users hands and eyes may be busy elsewhere, and the user may be busy avoiding the normal hazards of moving around, as well as engaging with real-world tasks. Fourthly, as devices diversify and proliferate, users increasingly face the need to make two or more de-

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

vices *interoperate for some ad-hoc purpose*. Even where each device has a well-designed user interface, this kind of task can be hard to arrange.

These four factors, singly and in combination, can cause difficulties in situations where the user does not know, cannot recall or cannot locate the commands needed to make the computer carry out a desired action. For mobile devices and their typical contexts of use, it is often inconvenient, impractical or too time-consuming to navigate to the appropriate screen or to otherwise search the space of available commands to effect the appropriate action.

In the case of context-aware devices, a final cluster of problems applies: there is no generally agreed uniform framework for applying contextual information. Context-aware systems tend to make incorrect guesses; and the user generally has little scope for correcting or capitalising on incorrect guesses [2]. This paper argues that Direct Combination has a part to play in addressing all of these problems, and has a useful role in any future framework for mobile and ubiquitous HCI.

2 Ambient Combination: Direct Combination Applied to Mobile HCI

The four general problem areas related to mobile user interaction noted above can be viewed in part as problems of *search*. That is to say, they all cause problems whenever a mobile device does not immediately afford the currently desired action, and users are consequently forced to navigate the interface, drill down, scroll or otherwise search the interface for the item needed to perform the intended action.

Direct Combination (DC) [3], previously applied to desktop computing, can often significantly reduce the degree of search required to operate user interfaces. In this paper, we argue that Direct Combination can be applied even more aptly to mobile computing and inter-device interactions. We will use the term *Ambient Combination* to indicate specifically the adaption, extension and application of DC principles to Mobile and Ubiquitous Computing [4]: however the most general term for the framework remains *Direct Combination*.

One convenient way to introduce Direct Combination in its Ambient Combination guise is by means of an imaginary interaction scenario featuring a magic wand. Binsted [5] has argued that imagined magic is a valuable source of inspiration when designing innovative forms of technology. We will use a hypothetical magic scenario to illustrate a crucial usability problem with magic wands (and other mobile devices). We will then illustrate how Direct Combination can address this, and other problems. Later in the paper, we will present interaction scenarios that we have investigated with a prototype implementation.

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

2.1 A Hypothetical Scenario: Harry and the DC Wand

Harry raised his wand towards the menacingly advancing Gator[1] *and tried to remember the spell for turning it into something harmless. It was no good; he just couldn't remember the right spell….*

Problems of this sort with magic wands are common in fiction and folklore. For example, the story of the Sorcerer's Apprentice deals with an inexperienced wizard who has memorised enough commands to start a process, but does not know, or cannot recall, the commands needed to control or stop it.

Harry suddenly remembered that this was one of the new Direct Combination Wands. He wouldn't need to recall the spell. Quickly looking around, Harry noticed a small stone on the floor. Pointing the wand at the Gator, Harry made the select *gesture and then made a second* select *gesture at the stone. A glowing list next to the wand presented the two available actions applicable to this particular* pair *of things:* propel the stone at the Gator *and turn* the Gator into stone. *Gratefully Harry activated the second command and the Gator froze into grey immobility.*

A key insight of Direct Combination is that if the user is allowed to indicate in advance *two or more interaction objects* involved in an intended action, the system can often use this information to constrain significantly the search space of possible commands. This allows the system to present to the user a space of focused relevant options to choose from, instead of the unrestricted space of commands. In an environment rich in objects of interest, users often know, or can recognize, what objects they want to use (a printer, a wallet, a car, a door, a document, etc) - but operations, apart from commonly used operations, tend to be relatively more abstract, and harder for people to recall. The contrast is between *recognition* (easy) and *recall* (hard). Given this insight, Direct Combination user interfaces give the user the freedom to specify the parts of commands in any order desired, for example *noun noun* (to be followed later by a verb), as well as the more conventional *noun verb*. At each stage, feedback is given on how this constrains the available options (examples follow later).

However, while this kind of pairwise interaction is sometimes invaluable, it is not always convenient. For example, sometimes the objects that would help to constrain the space of relevant actions are not to hand in the environment. Hence, Direct Combination requires that pairwise interaction always be made available in such as way that it does not get in the way of pre-existing interaction methods or practices. Direct Combination offers more ways of getting things done, but must never leave the user worse off than if DC was not provided - conventional methods should always be available. This can be illustrated by replaying Harry's imaginary encounter in more detail.

Harry raised his wand towards the menacingly advancing Gator and tried to remember a spell for making it harmless. Harry could see in glowing letters next to the wand the most common general spells, and category titles grouping all known spells, but he had no time to search this list. Pointing the wand at the Gator with the select *gesture, the list of general spells vanished, to be replaced by a list of the most common spells applicable only to Gators. The entire list of Gator-specific spells was now available - but Harry had no time to deal even with this extensive list. Remembering*

[1] An imaginary monster.

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

at last that this was a DC compliant wand, Harry quickly looked around. He couldn't see a pig, or a pig medallion, a mouse, a spider or even an ant, but he could see a stone on the floor and he made a second select gesture at the stone. The glowing list next to the wand shrunk to the two available actions applicable to this particular pair of things: propel the stone at the Gator *or turn* the Gator into stone. *Gratefully Harry activated the second command.*

The list of Gator-specific commands that appears when Harry selects the Gator alone is nothing new to DC enabled devices: many existing systems have such a feature. Even less remarkable is the fact that the DC wand would have allowed Harry to point at the Gator and then invoke the verb (spell) directly, if his memory had permitted. Although Direct Combination allows users to employ pairwise interaction *(noun noun ….)* whenever they wish, they are never forced to work in this way. A DC command processor must accept pairwise interaction, but also the more conventional ways of specifying commands such as *noun verb* and *verb arguments*. Indeed the Direct Combination principle of *Subsumption* (explained later) requires this freedom. DC also requires the system to provide appropriate feedback at all points that the user partially specifies the command, in order to show how choices so far constrain or suggest further choices. Finally, the power of Direct Combination is greatly enhanced [3] when the assignment of operations to pairs of objects need not be specified exhaustively by the designer, but can be propagated by inheritance from definitions in well-factored abstract classes, either held locally or in distributed fashion.

3 Scenarios illustrating Ambient Combination

We have informally illustrated two principles of Ambient Combination, pairwise interaction and Subsumption. Ambient Combination is most easily understood further by considering more detailed scenarios.

3.1 Technology required to support scenarios

Later in the paper we will detail the mix of technologies used in our implementation. However, for simplicity of exposition, for now we will describe the infrastructure in more general terms. Figure 1a shows the screen of a PDA. The PDA has a stylus for detailed work, but can also be operated one-handedly using buttons. Plugged into the system is a scanner for reading ID-tags indicating the *object* or *class* identity of devices and objects in the environment. The scanner allows the user to select items of interest in the environment, such as tagged documents, other PDAs, printers, book covers, telephones, merchandise, room nameplates, captioned photographs on the wall, etc., by scanning their IDs. The user may scan objects at a distance. In the context of Ambient Combination, selecting an object by scanning it is referred to as 'zapping'. The user always has full control of what is zapped and what is not zapped. Ideally, the scanner should form an integral part of the PDA, for example as a PCMCIA module. In our current prototypes we use 30-foot range barcode readers, but other technologies such as infrared scanners and RFID tags could serve equally well. The

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

PDAs are wireless networked, so that they can access object directories, retrieve representations of objects stored elsewhere, and consult DC servers, as explained explain later. When initially switched on, the PDA gives indirect access, via applications and hierarchical menus etc, to all of the commands available in the system, just as usual - as shown in Fig 1a.

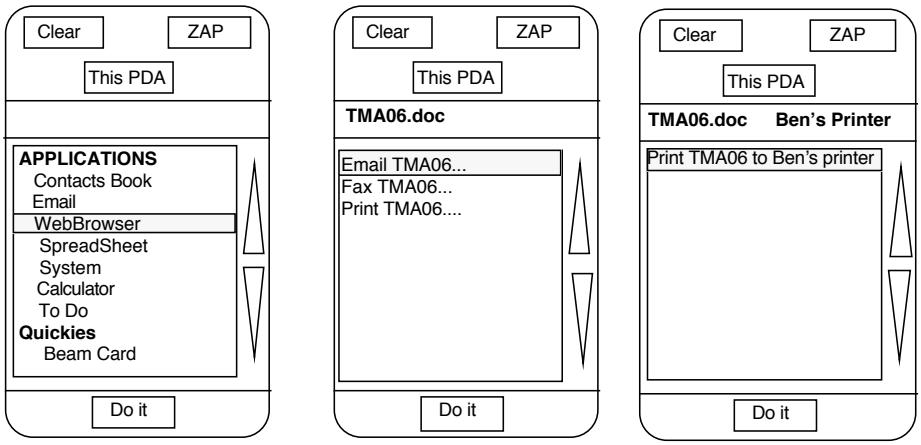

Figure 1a Figure 1b Figure 1c

Figure 1a shows no items zapped; Figure 1b shows document alone zapped; Figure 1c shows document and printer zapped, all in Scenario 1

3.2 Scope of scenarios

The primary purpose of the scenarios reported here was to help us prototype the interaction techniques and infrastructure necessary for AC, rather than to spotlight application areas where it would be most valuable. For an a priori attempt to characterise such situations and domains, see Section 7.

Scenario 1 (Document/Printer)
Anne walks into Ben's office one evening to use his coffee machine. While Anne leaves with her coffee in one hand, Ben mentions that he is reading a paper document that might interest her. Anne wants a printed copy to read later. To save the inconvenience, time and trouble of searching for the document on a server (Ben can't recall the pathname or even what the file is called) Anne uses her PDA to 'zap' the bar code on the document and then 'zaps' the printer in Ben's office. A suggested action is presented on the PDA (figure 1c). Anne presses the 'Do it' button on her PDA to accept this action. The document is printed on the printer. The suggested action is:

Print 'TMA06' to Ben's Printer (see figure 1c)

To understand this interaction fully, it is instructive to consider the state of the interface on Anne's PDA at three successive stages, as shown by figures 1a, 1b and 1c.

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

Stage 1: No items zapped
Figure 1a shows the options available before any items were zapped. The PDA simply displays a choice of application programs and some immediate actions - precisely what a PDA would offer if Direct Combination were not available.

Stage 2: Document alone zapped
Figure 1b shows the options available after the paper document alone has been zapped, with a default highlighted. The options are

Email 'TMA06.doc' ……
Fax ' TMA06.doc' …….
Print ' TMA06.doc'. …….

These actions are those that are relevant when all that is known is that the document has been selected. Any of these options could be chosen by the user straight away and an argument filled in manually or by more zapping. There is nothing new about such a facility - its equivalent on the desktop is called a "contextual menu". In this situation, the lack of novelty is a virtue: it means that, whenever only a single item is zapped, DC does not get in the way of the conventional way of doing things. If the user decides that the wrong item was zapped, the cancel button may be pressed to deselect that object, in which case the situation reverts to figure 1a.

Stage 2: Both items zapped
Figure 1c shows the options available after both items are zapped. This illustrates a canonical pairwise interaction, as spelt out at the beginning of the scenario. Zapping both the document and printer icon greatly reduces the search space of applicable actions, avoiding the need to drill down or otherwise search on the PDA or desktop machine. Note that the options do not depend on the document or printer alone - they depend on the ordered pair of object types that have been zapped - and would in general be different for a different pair of objects. Typically, there is still more than one option presented - though just one in this case. If the presented action(s) did not seem appropriate or if something had been zapped by mistake, the user need merely press *cancel* to get back to the state where only a single item, the document, was selected. At this point, the options available would revert to those in figure 1b. Anne might choose to select some other zappable object e.g. one of the wallcons (see below) to initiate another way of achieving her goal. Or Anne might press *cancel* again to revert to the situation where nothing was selected.

Scenario 2 (Document/Person)
Anne might want to have the document emailed to her rather than printed. Anne can achieve this using DC in many ways. For example, Anne could use her PDA to zap the document and then zap her own id badge, if she is wearing one. Her PDA then offers the following options, with the first as the default.

Email 'Report42.doc' to self
Fax 'Report42.doc' to self

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

Scenario 3 (PDA as Proxy for Person)
If Anne had not been wearing a badge, then she could have scanned the document and pressed the "this PDA" button on her PDA instead. This is equivalent to the PDA zapping itself. Amongst other roles, PDAs often act as proxies or representatives of their owner. Hence, the same options are presented in this case as when Anne scanned her own badge.

As a minor variant of scenario 1, Ben might not have a printer in his room, but might have 'wallcons' on his wall (physical or projected captioned wall icons) representing various commonly used entities or resources, such as colleagues, departments, companies, projects, printers, locations, etc. Anne could use her PDA to zap the bar code on the document and then zap the printer wallcon on Ben's office wall. The following two suggested actions are presented on the PDA.

Print Document 'Report42.doc' to Room 308 printer
Print Document 'Report42.doc' to Meeting Room printer

4 Principles of Direct Combination

We are now in a position to state the principles of Direct Combination, extended and adapted slightly to deal with mobile and ubiquitous interaction (See box below). We will now comment on the three principles stated in the box in turn.

Principles of Direct Combination

Principles of visibility and pairwise (and more generally n-fold) interaction
Subject to social, aesthetic, organisational and legal issues including privacy, ownership and security:

Every object of interest, both virtual and in the environment should be,
• visible (or perceptible),
• capable of a **range** of **useful** interactions with **any other** object of interest.

The available interactions between pairs (or n-tuples) of objects should be:
• diverse,
• tailored to each pair (or n-tuple) of object types.

Principle of Subsumption
DC interaction styles must be implemented in such a way that they are immediately available, but do not impede access to pre-existing interaction styles.

4.1 Visibility

Ambient Combination requires that as far as possible, every object of interest in a given environment (and in a user's computer) should be visible and capable of zapping (i.e. selection and identification using a pointing device). Issues of privacy, dignity, permission, safety and security must be met. The principle of *visibility* is borrowed from Direct Manipulation. However, many desktop interfaces that pay lip service to Direct Manipulation often fail to make objects of interest visible. In the area

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

of mobile and ubiquitous computing, despite the popular notion of invisible computers, with its welcome stress on aesthetic minimalism, visibility remains vital for many kinds of interaction, especially, for example, to avoid the possibility of uninvited context-aware interactions [2].

4.2 Pairwise Interaction (More Generally, N-fold Interaction)

In most conventional systems, commands may be constructed by the user only in a restricted, unidirectional manner. The initial noun followed by the verb must be selected before any arguments can be specified. The order of assembling the command would not matter if it were not for the fact that these restrictions have significant implications for the search strategies available to the user. The user may have a clear intention, but may be unable to recall, or may be ignorant of, the appropriate verb to achieve it. Unknowns for the user may include the name of a command, how it is categorised, whether it exists, or where to find it. Even once the user has selected the initial object of interest, this does not always greatly constrain the space of applicable commands, so the user may still have a large space of verbs to search. As already noted, if the user can indicate in advance *two or more* of the interaction objects involved in an intended command this typically greatly *constrains the search space of possible interactions*. This allows the system to present a space of focused relevant commands. If the choice of focused commands offered is inappropriate, the user may deselect items or select other items of interest until the system makes suggestions that are more appropriate, or that can be easily modified to be appropriate. The requirement that **every** object of interest be capable of diverse interactions with **any other** object of interest is a heuristic to encourage designers to seek meaningful pairwise interactions (and to endow objects of interest with suitable operations) rather than an absolute universal requirement. The reference to *virtual* objects refers to objects of interest represented in a computer. When using Direct Combination, it is possible to mix both virtual and real objects in a single interaction [6].

4.3 Subsumption

Subsumption ensures that users are never worse off with a system that includes DC than with a system that excludes it. For example, in the scenarios, all of the traditional ways of using a PDA are available when nothing is selected: hence users can still make use of the familiar *verb, arguments* pattern or the *noun, verb, arguments* pattern. DC may be viewed as, and is easily implemented as, *subsuming* these patterns of interaction. The principle of Subsumption is a common sense precaution, as it is not always convenient to use pairwise interaction every time. Thus DC does not impose anything on users: it simply provides greater freedom. In many ways it is surprising that we didn't identify and rid ourselves of this imposition earlier.

5 Context-Aware Interaction

In this section, we will present some scenarios that illustrate some ways in which Direct Combination can afford context-aware interactions. These scenarios are ana-

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

lysed in more detail in [6]. The scenarios contrast what happens when Anne zaps someone else's PDA with her PDA in four contrasting situations. In each case, Anne includes her own PDA in the interaction by using the 'this PDA' button noted in scenario 3.

Scenario 4
Anne meets a stranger, and zaps their PDA with her PDA (including her own PDA as described in scenario 3). The following option is offered.

Beam card to strange PDA

This scenario is hardly novel, though the underlying processing behind it is slightly unconventional, as we will discover in a moment. However, it makes a useful initial benchmark for the subsequent scenarios.

Scenario 5
Anne meets a research colleague Fred in the corridor. Anne zaps Fred's PDA with her PDA. The options offered are as follows.

Email 'Research draft' to Fred
Email 'Research notes' to Fred
Search folder 'ACDC 'for documents to email to Fred
Arrange meeting with Fred

Scenario 6
Anne meets a teaching colleague Tony in the corridor. Anne zaps Tony's PDA with her PDA. The options offered this time are as follows.

Email 'TMA04.doc' to Tony
Email 'TMA tutor notes' to Tony
Search folder 'TMA04 2002' for documents to email to Tony
Arrange meeting with Tony

In each of these scenarios, different interactions are offered, although the ordered pair of object types is the same each time - two PDAs. In scenarios 5 and 6, which have been implemented, the crucial difference is the identity of the owner of the zapped PDAs, and the nature of their work relationship with Anne, as previously recorded directly or indirectly by Anne, as we will outline below.

The technique currently used to support this simple contextual interaction is association matching: in each case, the direct combination of PDAs instantiates an operation where the initiating PDA will act as a disseminator of information. The key is to determine from the context how to prioritise or order the information to be offered for dissemination. The current mechanism is simple. Where the identity of the owner of the zapped PDA is available for the DC server to look up, activities associated with that owner can be retrieved (e.g. in this case particular research or teaching activities) and compared with associations noted against files or folders stored by, or accessible to, the initiating PDA. Documents whose associated activities or groups match the activities or groups associated with the owner of the zapped PDA are then prioritised for dissemination. In the case of the stranger, no previous relationships are recorded, so only options appropriate for strangers are offered. However, if electronic business cards were to be exchanged, and if either party went on to annotate the other's card at

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

a later point with discovered interests or roles in common (in a manner outlined below), more tailored options might be offered at a later encounter.

The next scenario, unlike the others, is not implemented in prototype. The necessary contextual inference used in this combination could be addressed by much the same technique as in the three previous scenarios, but seems better addressed by more powerful mechanisms, such as aspect-oriented programming and related techniques, as we are currently exploring. The next scenario, in contrast with the others, also involves *automatic sensing* of context (such as location) to complement the elective zapping we have considered exclusively so far (figure 2).

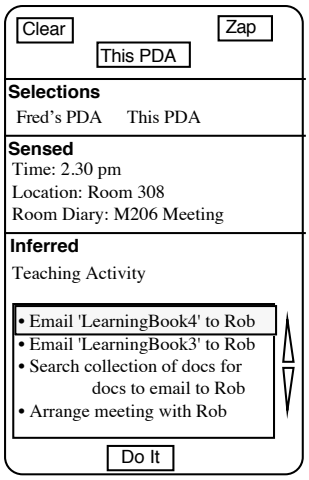

Figure 2. A hypothetical contextually-aware interaction (See Scenario 7)

Scenario 7

Anne goes to a course team meeting and sees Rob, with whom she collaborates on both teaching and research. Anne zaps Rob's PDA with her PDA. The options offered are as follows (see Figure 2).

Email 'LearningBook4' to Rob
Email 'LearningBook3' to Rob
Search folder 'Learning Book revisions' for documents to email to Rob
Arrange meeting with Rob

Scenario 7 involves zapping between two PDAs, but this time, the colleague has twin roles, as a research collaborator *and* as a teaching collaborator. The key idea is that inferences about activities associated with the current time and with the automatically sensed location should be used to order the likely relevance of the teaching and research roles in the given context, and hence to prioritise the collection of associated documents to be offered for dissemination. In this hypothetical scenario (figure 2), Anne's location is sensed to be in a meeting room, and the booking diary associated with the meeting room shows a meeting booked with a course team of which Anne is a member. Hence the context is used heuristically to prefer the teaching role as more likely to be relevant than the research role to the interaction. Until Anne actively zaps

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

something, these automatically sensed context objects (the location and the meeting - figure 2) are inactive in the DC system.

In all of the scenarios in this section, none of the interactions involve negotiation with the other PDA beyond extracting an object or class identity. Information from Anne's own diary, address book and desktop PC is used by Anne's DC server (see Section 6) to make the necessary inferences. In our prototype system, the DC server obtains knowledge about relevant relationships from earlier Direct Combination operations used to associate: work folders with activities (e.g. particular research projects or courses); individuals with groups; and groups with activities. These operations simulate Anne using desktop DC drag and drop operations [3] to maintain her desktop and address book. Similar mechanisms would apply, whatever sources of information were used.

The way in which contextual information is used in the implemented cases and in the hypothetical case is a little different from the way in such information is typically applied in conventional systems. We note four such differences. The first two are reflected in the implemented scenarios, the second two are not yet implemented, but follow from DC principles. Firstly, under DC, when a user selects two or more objects in the environment, this can be used to narrow down the set of intended actions, affording a substantial reduction in the search space. By applying contextual information at this late, relatively knowledge-rich stage, it can be applied in a highly targeted manner. This use in DC of contextual information at a knowledge-rich stage contrasts with many existing context-aware systems, which typically guess desired actions based on much less constrained evidence, such as automatically sensed information alone, or automatic interpretations of user's actions [2]. The second contrast is that, as is always the case in DC, the selection and execution of actions is left to the user, irrespective of whether contextual information is used or not: actions are not taken autonomously. A third difference is that under AC, context objects inferred from automatically sensed information, where context is used, are always made visible to the user (figure 2), who may opt to overrule the use of any particular item of context, or all of them. That is to say, the user always has the option of switching off the use of context in general, or particular sources of context, or explicitly discarding particular contextual inferences (e.g. inferences based on: location; calendar; address book; etc.) either until further notice, or for a given interaction. Finally, because DC creates such a knowledge rich environment, the user interface should always be able to afford the user fast ways to repair incorrect guesses on the part of the system by choosing alternatives to the presented objects. The way in which information about context is *detected* is no different in AC, but the way in which the context is *applied* appears to be different.

6 Architecture

For AC to be deployed in a given environment, various architectural elements need to be present. AC can be implemented in simple forms relatively easily. The environment needs to be made scannable in some way. We use optical barcode readers, including a thirty-foot range hand-held scanner, which are simple and effective, but

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

may also be viewed as proxies for other more sophisticated ID technologies. The user needs a feedback device such as a PDA screen. A DC server is needed (a system which, given references to two objects, computes the options to offer the user). This may be held locally (e.g. on PDAs or other devices) or on a central server. Distributed architectures are also possible. DC servers represent objects using well-factored class hierarchies [3]. In many cases, only the class of a scanned object need be known, not its state. However, in some cases, state may needed to help infer relevant operations (e.g. when context is used). For more information on how to build DC servers and create domain models, see [3].

6.1 Security

In a full AC system, scanned objects might not reveal their identity to all enquirers; objects might not reveal their state; and some networked DC servers might be available for specialised purposes only to those with the right *subscriptions* or *credentials*. Hence, much like other computer interactions, some AC interactions are likely to be freely available to all, others available only to those who have paid, or who have the correct accreditation. Similarly, some kinds of interaction might be freely available to all *within particular locations* such as railway stations, airports, libraries, schools, universities, places of work, or supermarkets. Yet other interactions might be available only to certain *kinds of people*, for example policemen, emergency workers, maintenance workers, with access policed by specialised wands or passwords.

7 When is DC useful?

Because of the principle of Subsumption, there is no need for pairwise or n-fold interaction to be useful all of the time. Even if it only helps some users with some interactions some of the time, DC can be useful, since it gives a simple uniform framework for these and other interactions without getting in the way. However, it would be useful to characterise the kinds of situations to which higher order DC interactions ($n \geq 2$) are especially suited. Empirical work is needed to answer this question fully, but since DC hinges in part on reduction of search, some a priori considerations can characterise such situations in part, as we now consider. Domains can be expected to be suited to higher order DC interactions provided they have: sufficient variety of objects, sufficient variety of operations, and provided that the operations are distributed between object type pairs. For AC (over and above Desktop DC) to be applicable, a scannable environment is needed where the objects of interest are visible in the environment. Given a suitable domain, users could be expected to benefit from DC in any of the following situations: the domain is unfamiliar; there are too many commands to search quickly; the user is focussed on the environment; minimal attention situations (where time and attention are scarce for searching the user interface, and are better focused on the environment); feedback bandwidth is limited; or the task focus is specifically on combinations of objects. More generally, DC is particularly useful for all kinds of data translation and interoperability. As noted earlier, much work with mo-

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

bile and ubiquitous devices involves ad-hoc tasks with novel combinations of unfamiliar resources. AC appears to be well suited to such situations.

8 Prototypes and their Limitations

The prototypes reported here were designed for purposes of proof of concept and for refining the DC mechanism. The architecture used in the most recent prototype can support some three-fold combinations, but it is not yet clear how well this would scale to n-fold combinations in general (i.e. when four or more objects are zapped in one interaction). The domain models for our prototype DC servers have been kept minimal. We did not connect the servers to the relevant services (e.g. print queue, etc) - the user was presented with dynamically instantiated textual stubs. One of our prototypes is designed to work on Casio PDAs via front ends coded in Smalltalk using a choice of onboard DC servers or communicating wirelessly to a DC server coded in Smalltalk running on laptop. However, for reasons of practicality, the demos reported here were carried out on laptops simulating PDAs with an onboard DC server. Tagging of objects in the environment was achieved using barcodes and USB hand-held bar-code scanners, including a 30-foot range scanner. The prototype server follows the Subsumption principle, and could offer relevant actions whether zero, one, or two objects were zapped.

9 Related Work

Using pointing devices to transfer information between devices (typically computers) is not new. For example, Pick-and-Drop [7, 8] is a pioneering extension of Drag & Drop used to copy data between multiple devices via passive pens with ids. Pick and Drop can be viewed as a natural extension of Drag & Drop to mobile and ubiquitous computing. In turn, the principle of Ambient Combination may be seen as an extension of Pick and Drop. AC offers potentially greater flexibility and expressiveness of interaction. The InfoStick [9] is an interaction device for inter-appliance computing. It may be used to pick up, and store, information items from a variety of devices, and then move them to other devices. The InfoStick may be viewed as offering a limited special case of Ambient Combination where the only available operations are *get* and *put*.

DataTiles [6] is an attractive tangible computing system using tagged transparent tiles placed on a flat display. Interactions between any two tiles are effected by physical adjacency, or by a pen gesture. The kind of interaction is determined by the tile types, although a pen gesture may be used to make this continuous or discrete in nature. DataTiles, like many tangible computing system, has the potential to be given greater flexibility and power, without loss of elegance, by applying Direct Combination in a variety of ways. For example: DC options could be visibly offered when two tiles were placed together or connected by pen. Alternatively, Subsumption would be better satisfied if DC interactions were invoked via a single additional pen gesture to invoke Direct Combination between two tiles. Or as a third alternative, a special

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

magic lens tile could be laid on top of inter-tile borders, to reveal and allow the selection of available interactions.

From one perspective, DC may be viewed as a novel way of exploiting a *relational* approach [10] to Tangible User Interface (TUI) design, *systematically* allowing the selection of multiple objects to determine dynamically bindings between objects and computational operations. However, although DC is a generally useful technique for TUIs, DC works beyond TUIs to allow mixed interactions between virtual and real objects [6].

DC relates strongly to Direct Manipulation, and as discussed in [3], parts of DC may be variously viewed as generalisations or specialisations of DM. However, since both DM and DC can be conceptualised at several levels, which do not mutually correspond, neither idea is a superset or subset of the other.

10 Conclusions

Ambient Combination (AC) is the adaption, extension and application of Direct Combination (DC) to Mobile, Ubiquitous, Tangible and Mixed Reality Computing. In many situations, Direct Combination can reduce the amount of search, time and attention required by users to carry out actions that they do not know how to perform quickly. DC appears useful where display real estate is limited or input devices are inconvenient. It appears useful in minimal attention situations, where time spent navigating a user interface is at a premium, and in situations where users need to make unfamiliar devices work with each other. DC lets the user combat the *combinatorial explosions* inherent in these circumstances by constructing defensive *combinatorial implosions* of relevant objects. Direct Combination allows the user to exploit objects in the environment to narrow down the range of interactions. It may be viewed as a new way of distributing the user interface in the environment. DC is not a replacement for other kinds of user interaction; it is a complement: standard interaction patterns may be subsumed as special cases. When pairwise combination is applicable, it can reduce the need for memorisation and interface navigation. DC allows users to employ recognition as opposed to recall. Direct Combination seems to offer a new approach to dealing with context-aware interactions. The strength of DC arises from the twin exploitation of user's perceptual, physical and spatial situatedness in the environment, and the organisation inherent in object hierarchies. Almost all of the *elements* of Direct Combination exist in fragmented or partial form in other systems: what is new is the simple uniform framework and approach that DC affords for dealing with challenging interactions, and simple abstract architectures to make this possible. We argue that DC has a useful role to play in any future framework for mobile and ubiquitous HCI.

Acknowledgements

Thanks to: Alistair Edwards, Claudia Eckert, Martin Stacey, Randy Smith, Bill Gaver, Benedict Heal. This paper is a shorter, revised version of [6].

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122

References

1. Pascoe, J. Ryan, N. and Morse, D. "Using While Moving: HCI Issues in Fieldwork Environments," ACM Transactions on Computer Human Interaction, vol. 7, pp. 417-437, 2000.
2. Erickson, Thomas (2002) Some Problems with the Notion of Context-Aware Computing. Communications of the ACM Feb 2002/Vol 45 No.2 pp102-104.
3. 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.
4. Weiser, M. "The Computer For the 21st-Century," Scientific American, vol. 265, pp. 66-75, 1991.
5. Binsted, Kim (2000) Sufficiently Advanced Technology: Using Magic to control the world. In Extended Abstracts of the ACM Conference on Human Factors and Computing Systems CHI 2000, pp 205-206 ISBN: 1-58113-216-6.
6. Holland, S., Morse, D.R., Gedenryd, H. (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, UK.
7. Rekimoto, J., Pick and Drop: A Direct Manipulation technique for multiple computer environments, In proceeedings of UIST '87 p.31-39.
8. Rekimoto, J., Ulmer B. and Oba, H. DataTiles: A modular platform for mixed physical and graphical interactions. In Proceedings of CHI 2001. 2001 p 269-276.
9. Kohtake, N., Rekimoto, J. and Anzai, Y. InfoStick: an interaction device for inter-appliance computing, Handheld and Ubiquitous Computing (HUC '99) 1999.
10. Ullmer, B., Ishii, H., (2000) Emerging Frameworks for Tangible User Interfaces. IBM Systems Journal 39 (3&4), 915-931.

This paper was published as: Holland, S., Morse, D.R., and Gedenryd, H. (2002). Direct Combination: a New User Interaction Principle for Mobile and Ubiquitous HCI. In: Paterno, Fabio ed. Human Computer Interaction with Mobile Devices. Lecture Notes in Computer Science (2411). Berlin: Springer, pp. 108–122