Four example of an application simulating an environment of context aware computing: proximate selection, automatic contextual reconfiguration, contextual information and commands, and context-triggered actions

**Proximate Selection :**

Proximate selection is a user interface technique where the located-objects that are nearby are emphasized or otherwise made easier to choose. In general, proximate selection involves entering two variables, the “locus” and the “selection.” However, of particular interest are user interfaces that automatically default the locus to the user’s current location.

There are at least three kinds of located-objects that are interesting to select using this technique. The ﬁrst kind is computer input and output devices that require co-location for use. This includes printers, displays, speakers, facsimiles, video cameras, thermostats, and so on. The second kind is the set of objects that you are already interacting with, and which need to be addressed by a software process. This includes people in the same room to whom you would like to “beam” a document. The third kind is the set of places one wants to ﬁnd out about: restaurants, night clubs, gas stations, and stores, or more generically, exits and entrances. Consider an electronic yellow pages directory that, instead of the “city” divisions of information, sorts represented businesses according to their distance from the reader.

Location information can be used to weight the choices of printers that are nearby. Figure 1. shows proximate selection dialogs for printers using three columns: the name of the printer, the location, and a distance from the user. One interface issue is how to navigate dialogs that contain this additional location information. For example, should dialogs use the familiar alphabetical ordering by name or should they be ordered by location. Shown here are (a) alphabetically ordering by name; (b) ordered by proximity; (c) alphabetical with nearby printers emphasized; (d) alphabetical with selections scaled by proximity, something like a perspective view.

Another factor that proximate selection interfaces must take into account is bandwidth requirements. Presenting information that changes, either due to the user moving or the contents of the dialog changing (e.g. other people moving) will cause update network trafﬁc. One approach is to view location information with more or less precision based on the situation. The interfaces in Table 1. are ﬁne-grained – the distance column requires updating for each change in location of the locus. In contrast a coarser-grained view of the same information might show a zone rather than a distance. Driving around town with such a dialog would, for example, change only when the viewer, or the objects in the selection dialog, crossed the city limits.

Proximate selection may also be used to choose virtual objects. Using the PARCTAB voting application, users select previously created ballots either alphabetically or by the current location. This use of proximate selection is helpful when ballots are referenced at particular locations – e.g., voting on what snacks to have at High Tea – or when you are meeting with a group that has just created a ballot.

User interfaces for proximate selection pose some challenges. For example, how can a UI display both alphabetical and proximity information simultaneously. Map imagery may provide a good UI metaphor. Since proximate selection may occur on a mobile host, the UI techniques developed must take into account device capabilities such as screen real-estate and communication bandwidth.