1 INTRODUCTION

Wearable computers will strongly impact processes on the construction site in the future. Having information on construction elements, specifications, check lists or even immediate contact to an expert on site traveling time and resources can be saved. Moreover detailed, structured information can be provided on the construction site. But not only to provide information, also to collect data on site, wearable computers will have a great impact. Progress Monitoring and inspection tasks for instance can be made much more efficient by using wearable computers (Garrett 2000).

Having brought the computer on site helps little if the process of data input on site does not address the special requirements the construction site imposes. A keyboard will not be an option for a computer that is used on site, as the user would have to set down the computer to put in data. This paper discusses how wearable computers for construction sites can be equipped with functionalities that allow the user to fulfill the desired tasks on site.

2 WEARABLE COMPUTERS

Wearable computers are an emerging technology that will strongly influence data procession and communication processes on site.

Many people already take advantage of mobile computers such as notebook computers. It is a fact, that these computers have contributed to a higher productivity, because, the notebook computer is available to the user more often than a desktop computer. The use of a notebook computer is not an option for the rough environment of a construction site. Notebook computers have to be set down to input and retrieve data. A computer that can be used on site has to be robust, small in size and unobtrusive to the user.

There are two major categories of computers that meet this criteria. Firstly there are the emerging palm top computers. There is a growing amount of software available for palm top computers. AutoDesk has come up with a program that displays drawings on the palmtop. Nevertheless palmtop computers have a fairly small screen size and are therefore only to a limited degree usable to depict drawings or complex user dialogs. For more complex dialogs and to display drawings, a new generation of WIntel based wearable computers can be used.
These computers, on which this paper sets the focus, can be worn on the belt or a vest and therefore meet the requirements, the site context imposes. The interface to the user can be a hand held or a head worn display. Even though there are hand held keyboards available for these computers, efficient data input will only be possible by using the hand held touch screen and a speech interface. Therefore the user interface of the program to the user is of crucial importance.

There are applications available, that add a speech functionality to standard software applications. Tests have shown, that this added speech functionality does not work very well, in many cases. Applications that use a speech functionality that is embedded in the context of the program will work much better.

The next section will discuss, how this embedded speech functionality can be build in a software.

3 DELIVERING A SPEECH FUNCTIONALITY TO A PROGRAM

For the project, this paper is based on, a speech engine had to be linked to a java program.

A speech engine recognizes spoken text and synthesizes speech. This functionality had to be implemented in a program, that enables the user to access information and put in data on site.

Java provides a Speech API, java.speech, that connects a Java program to external speech processing software. In our case we used IBM's ViaVoice speech recognition, which turned out to be fairly robust and accurate for this particular application. The java.speech API provides a speech engine that has two major sub interfaces: Recognizer and Synthesizer. Whereas the Recognizer transforms spoken text to written text, the Synthesizer generates speech text from written text. Both features are incorporated in ICMMS.

To obtain the text of spoken words from the Recognizer, a grammar had to be defined.

The Java Speech API supports two types of grammar that can be applied by a speech engine. A Rule Grammar is a user defined grammar that is used to recognize key words and phrases that, for instance control a program. The second type of grammar is the dictation grammar, which is based on a predefined grammar provided by the speech engine's vendor. A Rule Grammar is much more accurate in recognizing spoken text than a dictation grammar as it selects the recognized words and phrases from a limited source of predefined grammar rules. Rules are sets of tokens that define the source from which the speech engine can choose phrases. Only if the spoken text coincides with one of the rules, the engine will recognize it. It is possible to use more than one grammar at a time at a speech engine.

There are different ways to create new Rule Grammars. The easiest way probably is to create a text source that complies with the Java Speech Grammar Format (JSGF) and use the loadJSGF method of the Recognizer to load the text into the Recognizer. Storing the JSGF text in a file that is accessed by the loadJSGF method, the file could have the following content:

```
grammar construction;
<job> = architect{architect}|engineer{engineer}|owner{owner};
public <role> = I am the {job} <job>
public <bye> = good bye {bye};
public <pile> = The pile has to be
moved {dir} <direction>
public <direction> =
up{up}|down{down}|left{left}|right{right};
```

It is easy to recognize the pattern on which the JSGF is based. The single phrases can be joined together. A possible phrase that the recognizer would understand would be for instance "The pile has to be moved left". The words inside the brackets are tags that can be recognized by the speech engine when recognizing the specific phrase.

The second, more difficult way to create a Rule Grammar is to create it from within the code. The advantage of this method is that the grammar can be altered or appended while the program is running.
A Rule Grammar contains a collection of Rule Tokens. Rule Tokens can consist of other Rule Tokens that are linked to each other by Rule Alternatives or Rule Sequences, which are children of the java.Speech API RuleToken class (Java Speech).

Our application uses a Rule Grammar to recognize speech commands that control the program, do data input and navigate through the drawing. When the user selects a certain element on the drawing, the Rule Grammar is updated, to ensure that the commands that are specific to the selected element are available in the Rule Grammar.

The Recognizer, which uses a Rule Grammar very much contributes to the high level of performance of the speech interface of our application.

Although the Rule Grammar delivers much more accurate results for command recognition, a Dictation Grammar is also needed to recognize comments the user may wish to make on the selected element. Words and phrases can also be added to a dictation grammar, but as the source from which text can be chosen is much broader in scope, the accuracy and speed of speech recognition is lower than when using a Rule Grammar.

4 NAVIGATING THROUGH A DRAWING BY USING SPEECH

Another important use of a speech interface for the application is to provide a functionality to navigate through a drawing using only speech commands, such as:
* Move left
* Move right
* Move up
* Move down

To define the relative positions of the elements in a drawing, each element has specific points that represent the location of the element. A good point to use is the centroid of the element. The centroid can easily be calculated and is probably very close to the point the user naturally associates with the location of the element.

Having reduced the elements to centroid points, all centroid points located in a certain direction relative to the reference point (e.g. right, up), can be determined. If the elements are sorted, the one centroid point that is the closest to the reference centroid point can be determined.

![Figure 3, Selection process of an element, using speech](image)

Often the centroid point of an element that is the closest to the reference point, with respect to the desired direction, is not the one the user naturally associates with requested movement. For instance element 3 of the drawing shown in Figure 3 is the closest element to the reference element R if the requested direction is "right". Nevertheless, the user naturally associates element 4 with the direction right, even though it is not the closest element to the reference element to the right.

For this reason an additional filter condition has been introduced, which only considers elements, that are within the triangle that is defined by the broken lines of the desired direction.

For example the condition for a "Move Right" request would be:
\[
\text{diff x} > \text{diff y}
\]

Applying this condition, element 4 will be selected. For a Move Up request, the filter condition would be:
\[
\text{diff x} < \text{diff y}
\]

Element 3 will then be selected. To determine the elements that are located in a certain direction relative to the reference element, the program takes advantage of SQL requests, which return only the elements that meet the specific conditions of the SQL request. For example to get a list of elements that are right from the reference element the following java and SQL code can be used:

```java
String queryStr;

if(direction=="right"){
    queryStr = "SELECT * FROM tabElements WHERE centroidX>" + (currentX+0.000001) + " ORDER BY centroidX";
} else {
    queryStr = "SELECT * FROM tabElements WHERE centroidX<" + (currentX-0.000001) + " ORDER BY centroidX";
}
```

```sql
SELECT * FROM tabElements WHERE centroidX>" + (currentX+0.000001) + " ORDER BY centroidX"
```
In this example, currentX and currentY represent the coordinates of the reference element. The variable queryStr contains the String that represents the SQL request to the database. All elements should be returned from the table tabElements that have a X - value of their centroid that is greater than the X - value of the reference element which is represented by the variable currentX. By applying the ORDER BY centroidX literal, the elements in the returned ResultSet will be in the order of their distance to the reference element with respect to the X - Value.

The following code shows how the filter condition is applied to the ResultSet which is returned from the database.

```java
ResultSet rs;
Vector reply;
...
if((direction=="right")|| (direction=="left")){
    if(diffY <= diffX)
        {reply.add(new Integer(
            rs.getInt("elementID") )); }
}
```

The Vector reply will contain the IDs of the elements that are to the right of the reference element and meet the filter condition which says that the distance of the reference element to the desired element has to be smaller with respect to the y - Value than the distance with respect to the x - Value. The first element that meets the filter condition is the new selected element. The quoted code is embedded in a loop that runs until the first element meets the filter condition or the ResultSet has run out of elements.

This, of course, is a very simple algorithm to select the desired element. It will take several steps to move the focused element from the upper left part to the lower right corner of a drawing.

There are also alternative algorithms that could be applied:

**Multiple Step Algorithm:**
1. Select a direction ("move right")
2. Tell the computer the number of elements to move to the desired direction

This algorithm is easy to implement as it just skips the first n element IDs that are returned from the SQL request and meet the filter condition.

The algorithm also would be very efficient to select elements that are located in a regular order (e.g. grid of columns).

**Number Selection Algorithm:**
1. Assign numbers to the elements that surround the reference element
2. Tell the computer the number of the desired element

Using this algorithm would make it easier to select elements that are not clearly in one of the four major directions from the reference element. Also navigation on layouts that have a rather random distribution of elements would be improved as it takes only one step to select the desired element. The disadvantage would be that the numbers of the elements would have to be depicted on the drawing. On the one hand, the numbers should be large enough to be readable by the user, while on the other hand the elements should not be covered by the numbers. This issue is especially important for the wearable computer which has a limited screen size.

**Scale Dependent Selection Algorithm:**
1. define a point P on the desired side of the reference element that depends on the current scale of the drawing on the canvas, and
2. select the element that is closest to point P

This method will work best in conjunction with a powerful zoom function. The selection depends on the current scroll and zoom settings. As scroll and zoom settings have to be adjusted in many cases anyway, there is only one more step ("select the direction") to select the desired element. The disadvantage of this algorithm is that scroll and zoom settings have to be very accurate to select the right element. Nevertheless, it is a good algorithm to move the selected element over larger distances across a drawing.

The most efficient way to navigate through a drawing is very much dependent on the specific characteristics of the drawing. All algorithm have advantages and disadvantages. Powerful speech navigation strategies therefore should allow users to select the most suitable navigation algorithm.

5 APPLICATIONS OF SPEECH CONTROLLED WEARABLE COMPUTERS

Progress Monitoring for construction projects is becoming increasingly important, as owners demand shorter periods for the delivery of their projects. Construction companies have great interest in knowing where problems can occur and if the actual construction progress is eventually behind the planned construction progress.

Traditionally, Progress Monitoring is done by marking the actual progress on the construction site in a layout. Back in the office the person who does progress monitoring transfers the visual information from the layout in a progress figure and compiles charts that depict the construction progress over time.

This process can be very time consuming and has to be repeated several times during the duration of a construction project. For this reason, Progress Monitoring is often enough neglected, which may result in unrecognized delays and coordination problems of the construction project.

In a joined project of Carnegie Mellon University and Technische Universität Dresden, a prototype of
a wearable computer based Progress Monitoring System ICMMS (Integrated Contract Management and Monitoring System) was developed. ICMMS supports the processes of collecting progress data on site and the procession of this data in the office. The screen of the wearable computer that is used on site shows a drawing of the construction project. To input progress, the user selects a building element, that is depicted on the screen, selects an activity that is carried out at this particular element and puts in a progress figure. This update process can be done by using a stylus and a touch screen. In many cases, where hands free operation of the wearable computer is required, the user is reduced to input construction progress by using the speech interface, which incorporates the methods and components that have been discussed in this paper.

Figure 4, Flat Panel Display of Xybernaut Wearable computer

Doing Progress Monitoring with unobtrusive, lightweight wearable computers, that have a user interface, that is tailored for the specific task and environment, data collection and procession time can be saved. ICMMS, that provides such an interface, can make information about the status and problems of a construction site available much earlier than doing Progress Monitoring traditionally by hand (Reinhardt 2000).

6 SUMMARY AND FUTURE DIRECTIONS

Mobile computing has shown in many fields, how productivity can be increased, if data is put in and retrieved from the computer in the place where it is needed. Even though this is also true for construction industry in general, computers are rarely used on construction sites.

The emerging product group of wearable computers and also palm top computers can help to make the advantages of mobile computing also available on the construction site. Nevertheless, the software that is run on these wearable computers has to respond to the specific context, the wearable computers are used in. For applications that depict a drawing on the screen of a hand held display, special ways of navigation through a drawing have been examined. Besides identifying drawing elements by clicking on them with a stylus, speech can be used as an interface to interact with the drawing. Commercially available speech recognition tools can be linked to new applications. It is important, that the designed application is structured in a way that follows the logic of a speech dialog between user and computer. Navigating through a drawing is of special interest for research, as efficient selection and control processes for the user interaction with a drawing can become very complex.

As the introduction of wearable computers in new fields, such as construction sites, advances, the topic of “keyboard - less” interaction with the computer will gain importance.

Of course the navigation methods stated in this paper are a first step in the field of speech controlled navigation through drawings. Further research will be necessary to improve these methods. It would be also interesting to explore, in how far speech can be used to navigate through a virtual 3D model of a building. It is also conceivable to support the navigation process by external tools, such as a GPS receiver. The system could automatically select the building element on the screen at which the user on site is standing.

Further fields of research are how peripherals, such as digital cameras can be linked to wearable computers and how the communication process between office and construction site can take advantage of this.

Nevertheless the main objective of further steps should be, to improve the interaction between user and computer in different field contexts.

REFERENCES


Progress Monitoring with wearable Computers. available as: http://www.ce.cmu.edu/~janr/icmms/icmms.