The preliminary design of a wearable computer for supporting Construction Progress Monitoring

Jan Reinhardt, TU - Dresden
Prof. James H. Garrett,Jr., Carnegie Mellon University
Prof. Raimar J. Scherer, TU - Dresden

1 Introduction

Progress monitoring has become more and more important as owners have increasingly demanded shorter times for the delivery of their projects. This trend is even more evident in high technology industries, such as the computer industry and the chemical industry. Fast changing markets, such as the computer industry, force companies to have to build new facilities quickly.
Progress monitoring enforces the requirements stated in a construction contract by tracking construction time, ensuring that milestones are met, and coordinating activities on site. An optimized schedule for a construction project is only able to have an effect if it is frequently checked against the actual construction progress on site and changes are made if the actual schedule does not agree with that planned.

To make a statement about construction progress, the status of a building has to be determined and monitored over a period of time. Depicting the construction progress in a diagram over time, statements can be made about the anticipated completion of the project and delays and problems in certain areas. Having this information, measures can be taken to efficiently "catch up" on the schedule of the project.

At many construction projects, progress monitoring is very much neglected. Often the impact of a delay of one activity will become clear at the start of a later activity, which cannot be carried out as its preconditions are not fulfilled. It takes much more effort and cost to catch up on the schedule. One reason construction progress is not traced more closely is that it still takes a large amount of effort to collect status and progress data on site and to process it.

New technologies, such as wearable computers, speech recognition, touch screens and wireless networks could help to move electronic data processing to the construction site. Progress monitoring could very much take advantage of this move, as several intermediate steps of processing progress data can be made unnecessary. The processing of progress data could be entirely done by computers, which means that data for supporting decisions can be made available at the moment the construction progress is measured.
It is desirable to have a tool that enables the progress monitoring engineer to measure the construction progress efficiently. Such a system would have to respond to the specific requirements imposed by a construction site on the usage of a computer. A wearable computer, as shown in Figure 1, could be used to input progress data on site. The wearable computer has the advantage over a notebook computer, because the user does not have to sit down or set the computer down to operate it. Doing all data input via a touch screen or a speech recognition interface, the keyboard would be unneeded. Such an interface will allow progress monitoring engineers to move about the construction site and record progress information.

Figure 1, Wearable Computer Xybernaut MA - IV

This paper discusses a preliminary design of a wearable computer for supporting progress monitoring.

2 Progress Monitoring

Marking construction progress in a drawing and translating this data into a progress figure is not a highly efficient progress monitoring system. Using wearable computer components could make the process of transferring progress data from a drawing into progress figures much faster, as the steps of marking the construction progress in a drawing and calculating a progress figure would be made unnecessary. The visible construction progress on site could be directly input into the computer on site. The computer could calculate the updated changes and give the progress monitoring engineer instant feedback concerning the impact the new progress of an updated element has on the progress of the single activities and the overall project. This feedback will also give the progress engineer a good idea of the importance of the single elements for the completion of the overall project.

Having accurate and reliable progress figures for a project, weekly progress reports could be compiled. A progress report could consist of diagrams that show the overall completion and also the completion of subparts of the project. Drawings could provide information about where problems are located and what elements are finished. The link between the schedule, status data and the locations of the individual activities within a construction project can help to make complex construction processes easier to manage. Elements that are ready for the next activity could be automatically highlighted in a drawing and show the construction workers where to continue work. These are just a few examples of other processes that could make use of the progress data gathered on site. It is also conceivable to connect the wearable computer to a digital camera. Pictures taken on site could be assigned to elements and the activities to which they are related. This feature, combined with a speech controlled comment editor, could be used to create work orders or site observation reports to subcontractors on site.

3 The ICMMS architecture

The system we developed in the project described in this paper is named Integrated Contract Management and Monitoring System (ICMMS). Due to the fact that construction sites often are not located where the data gathered on site will be processed, ICMMS consists of several components that are connected by a wide area network. It is also conceivable that several clients have to access shared data. ICMMS therefore uses a central database that stores all data, which is a crucial feature of ICMMS to enforce the consistency of data. As shown in Figure 2 there are two types of clients. The Field Client is a program that has been designed to support the efficient input and access of data on site and is run on a wearable computer. The Field Client has to respond to special requirements of a wearable computer that only supports input done via touch screen or speech recognition. The other client, the Office Client, is used to set up progress monitoring systems and process and evaluate data that has been collected by the Field Client. The Office Client has been designed to be run on an ordinary desktop PC. The connection between the wearable computer and the central ICMMS database can be provided by a wireless network connection. Although recent wireless network technology is very powerful and reliable, it might be difficult to establish a connection between a wearable computer that is located in a basement of a building and the central database server that is located in the construction office. For these cases, ICMMS provides an option to run the Field Client offline. The Field Client then connects to a local database on the wearable computer. As soon as the central ICMMS database is accessible

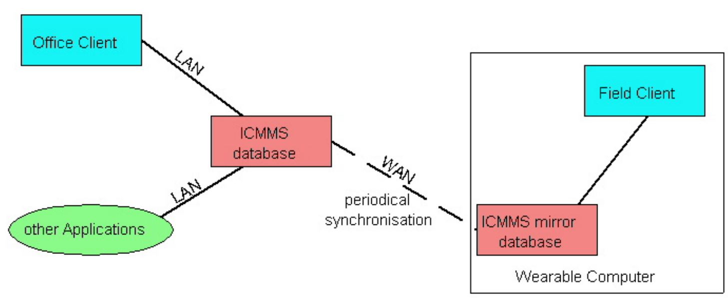

Figure 2, Overall architecture of ICMMS

again, via a wireless or a conventional network connection, the local database on the wearable computer synchronizes with the central ICMMS database. Often construction sites are far away from the construction company's offices. This raises the need to design the connection between the central database and the Field Client as a Wide Area Network (WAN) connection, whereas it will be sufficient to connect the Office Client and other office applications via a Local Area Network (LAN).

4 Field Client Functionality

One of the main objectives of the project described in this paper was to design and implement a first version of a Field Client. The main task of the Field Client is to provide an interface to progress monitoring engineers, that allows them to input and access data efficiently under site conditions. Therefore, the Field Client should have a drawing of the project that is, on the one hand, accurate enough to allow the progress engineer to identify building elements, and on the other hand as abstract as possible to enable the user to select elements on the drawing by picking them with a stylus (Figure 3). This first version of the Field Client uses a 2D graphical interface, which appears to be sufficient for the particular task of progress data input on site. Using a more sophisticated 3D graphical interface could make the entire process of picking elements and navigating through a drawing more difficult and therefore inefficient to be done on a wearable computer. The geometry data of the depicted elements is read from a dxf - file into the database. Reading this data from the database the Field Client draws the elements as filled polylines, which is accurate enough to be recognized by the user. In future versions of ICMMS, a more elaborated concept may replace this approach.

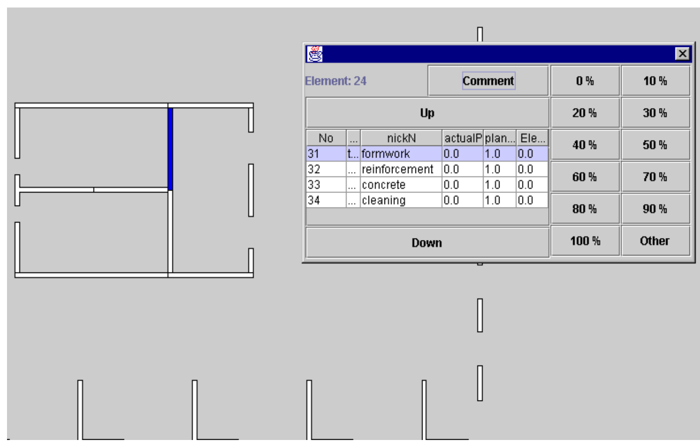

Figure 3, User interface for progress input

Upon selecting an element on the drawing, information should be made available on the selected element. The progress monitoring engineer needs to be provided a list of the activities that are carried out on the specific element. Furthermore it would be desirable to have information on the actual and planned status of the activities that are related to the element. It should also be possible to make a comment that relates to the selected element, an activity or to the part of an activity that is carried out on the specific element. Moreover, the interface should have buttons that enable the progress monitoring engineer to input progress information just by picking the buttons with a stylus. These features have been implemented in the first version of the ICMMS Field Client.

To update progress, the progress monitoring engineer selects an activity that is to be carried out on the selected element and picks the button that indicates the desired progress to be recorded.

The process of updating progress can also be done via an implemented speech recognition interface. A typical dialog between the progress engineer and the computer would be the following:

User: "move up" (Navigation to the desired element)
User: "move left"
User: "what are activities of the element?"
Computer: "Activities are excavation and foundation and backfill"
User: "excavation"

Computer: "excavation selected"
User: "activity 100 % complete"
Computer: "update done"

To support the process of identifying an element and navigating to it, ICMMS has a zoom and a scroll function that can be controlled via speech input and via the touch screen. Upon clicking with the stylus on the background of the drawing, a dialog box appears that has the 2 buttons to zoom in and zoom out on a drawing. Scroll functionality is provided by the scroll bars of the drawing. It is important for the ease of use of the system to access scroll and zoom functions quickly. The limited size of the screen of the wearable computer makes it necessary to take special care of the accessibility of all elements, whatever scale they are, in a drawing.

Figure 4, Input of progress data on site

The Field Client not only collects data, but it also provides information to the progress engineer. Query functions are provided to mark the elements that meet desired query criteria, such as activities that are late, finished or underway or are on the critical path of the schedule.
As all data that is input to the Field Client is tagged with the date and time of the input, the query function can also mark elements that have not been updated during a certain period of time. Querying the database for those elements that are not finished, not started, not due to be under way at the present time and have not been updated for a certain time, the progress monitoring engineer can confine his activities to update the elements that fulfill this criteria but still has the information available to obtain accurate progress figures.

Of course, there are also things happening on construction sites, that cannot be expressed by a progress figure. For these cases, ICMMS provides a function to make a comment that is assigned to an activity, an element or a part of an activity that is performed at a certain element.

5 Field Client Architecture

The main component of the Field Client is the drawing that depicts the elements of a construction project. Upon clicking on one of these elements, a dialog box appears that shows all available information that is related to the specific element to the progress monitoring engineer. The progress monitoring engineer can now edit or add progress data and make text comments. All this input will directly be transferred to the connected database. In this first version of the Field Client, all data is stored in one database which can be either the local database on the wearable computer or the central ICMMS database. Upon data input, the new data is directly transferred to the database.

The speech recognition interface is one of the features that makes it easy to input data to a Field Client. In many cases, it is likely that a head worn display in conjunction with a microphone and speaker head set will be used. In these cases all data input has to be done via the speech interface. All data that is accessed via the mentioned dialog box also can be retrieved and put in via speech recognition. The speech engine, which is a central element of the Field Client class structure, listens to the speech commands of the user. All commands that are recognized and comply with the predefined grammar of the speech engine will be transferred to a subordinate class that parses the recognized speech commands and triggers the appropriate functions. This intermediate class therefore has to be linked to the speech engine on the one hand, but also to the classes that specifically address the desired functions on the other hand.

6 The Field Client Speech Interface

A speech engine that recognizes spoken text and synthesizes speech had to be implemented in the Field Client. Java, the language for ICMMS, provides a Speech API, java.speech, that connects a Java program to external speech processing software. ICMMS uses 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 RuleGrammar 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 RuleGrammar 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 will the engine recognize it. It is possible to use more than one grammar at a time in a speech engine.

ICMMS uses a RuleGrammar to recognize speech commands that control the program, do data input and navigate through the drawing. When the user selects a certain element the RuleGrammar is updated, to ensure that the names of the activities of this particular element are added to the RuleGrammar. The Recognizer, which uses a RuleGrammar, contributes to the high level of performance of the speech interface of ICMMS.

Although the RuleGrammar delivers much more accurate results for command recognition, a dictation grammar is also needed to recognize comments the user may wish to make on activities or elements. Words and phrases can also be added to a dictation grammar, but as the source from which text can be chosen becomes broader in scope, the accuracy and speed of speech recognition becomes lower than when using a RuleGrammar.
Another important use of a speech interface for the Field Client is to provide a way 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 of a drawing, each element has specific points that represent the location of the element.

7 Summary and future directions

ICMMS is a prototype of a Progress Monitoring System. It has been designed to enhance the efficiency of progress monitoring for construction processes that, comprises the set up of the monitoring system as well as the collection, processing and evaluation of progress data. Tests of ICMMS on a real project had not been performed at the time of writing. As the program is based on techniques and methods that are commonly used by construction companies for large scale projects, there are no entirely new processes and methods involved in the ICMMS concept. Hence, we would be testing the effectiveness of the device for supporting existing processes.

The research especially focused on how the process of collecting progress data on site could be made more efficient. New technologies, such as wearable computers with touch screens, wireless networks and speech recognition software were used to build a prototype device that enables the progress monitoring engineer to measure progress on site efficiently. Data input can be done via the touch screen of the wearable computer or via speech commands. In some cases, where hands free operation of the computer is desirable, the entire process of updating progress can be done by using speech commands. The graphical interface of the Field Client has been designed to respond to the special requirements the field context and the wearable computer impose on data input. Using a wireless connection between the wearable computer and the central server on which all data is stored, the latest update of the construction progress will be available to all users at the very moment when the progress is updated. If it is not possible to establish an online connection between the server and the wearable computer, the Field Client can also work offline. The progress data on the server can then be updated by synchronizing the database on the server with the Field Clients. Moreover, ICMMS supports multiple users and remote data access, which qualifies the system to be used for large scale projects where several persons are monitoring progress. Due to the network structure of ICMMS, it is possible to use only one database on which all data will be stored. This is a precondition for the consistency of the data.

The ICMMS system provides a basic platform for using computers on construction sites. New features and modules could be added to the system to expand ICMMS to a Project Management system that also comprises taking and handling of photos, the compilation of invoices, ordering of materials, etc. Furthermore, the possibilities of using 3D models to depict building geometry should be investigated. Java's VRML API provides 3D modeling support that can lead the user through a virtual model of a building. Using a GPS receiver and a compass, the wearable computer could possibly calculate the position and orientation of the progress engineer, which means that all navigation will be done by the program.

The developed software should help to improve the performance of progress monitoring as it reduces progress monitoring to setting up a system and collecting progress data on site. The steps of transferring progress data from a drawing to progress figures and creating progress reports are done by ICMMS. The main benefit of an automated progress monitoring system like ICMMS is the savings in processing time for progress reports. This will enable construction managers to react earlier to delays and problems in the construction process, as data supporting decisions will be available immediately after the construction progress has been updated. Establishing a computer based progress monitoring system will help to make the construction process more transparent for the contractor, the owner and also subcontractors.

8 References

1. Progress Monitoring with wearable computers, available as http://www.ce.cmu.edu/~janr/icmms/icmms.html
2. Java speech Application Programming Interface description, available as http://java.sun.com/products/java-media/speech/
3. JavaTM Speech API Programmer's Guide, available as http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/index.html
4. Bhamidipati, Kishore: “SQL programmer's reference”, Berkeley : Oxborne/McGraw-Hill, c1998
5. C.J. Date, Hugh Darwen: “A guide to the SQL standard”, 4th ed., Reading, Mass. : Addison-Wesley, c1997