

Theme:

Title: **Mobile Devices for Engineering Analysis**Author(s): 1) Matthew Pilgrim, 2) Dino Bouchlaghem, 3) Michael Holmes,
4) Dennis LovedayInstitution(s): 1) Centre for Innovative Construction Engineering, Loughborough University
2 & 4) Loughborough University, Civil and Building Engineering
3) Arup Research & Development

E-mail(s): Matthew.pilgrim@arup.com

Abstract: *Handheld computers are now sufficiently small to be carried casually and as such it is evermore common to find engineers with these devices. This paper explores the possibility of providing remote accesses to typically office bound analytical software from these devices. With particular emphasis on the design of both input and output human computer interaction.*Keywords: *HCI, Mobile, PDA (Personal Digital Assistant), Engineering, Analysis***Introduction**

Engineers rely heavily on analytical software to assess building design; this must be performed at an appropriate point in the project and to the required level of accuracy. This can range from final design calculations at design completion to the rapid assessment of design options at the conceptual stage. Final design calculations typically rely on a broad spectrum of resources (such as analytical tools, drawings, design specifications etc.) that are only available within the designers' normal working environment. Rapid assessment of design options may however be carried out using the same analytical tools but based on reduced input data often supplemented with a number of assumptions. The rapid assessment of design options (at the conceptual stage) may therefore be performed when the engineer is away from their normal place of work, if they have access to their analytical tools. Facilitating the use of these tools at the conceptual stage of design is important as it is seen as the most innovative stage of the process. Decisions made at this stage are likely to have the greatest influence on the final outcome [1]. However at present computing devices capable of such calculations are still too large to be carried casually and therefore the tools remain unused in meetings and workshops where prior provision for their use hasn't been made. As an increasing number of staff are adopting handheld computers (Personal Digital Assistants or PDAs) a study was undertaken to examine if they could be used to perform engineering calculations. This paper outlines a project in which recently established mobile computing and communication technology was used to develop a system for remotely entering data, running an advanced façade analysis tool and reviewing the results.

Related Work

A brief study of Human Computer Interaction (HCI) and mobile devices is followed by a review of data representation techniques relevant to small displays.

Human Computer Interaction & Mobile Devices

William Mitchell identified wireless mobile devices as one of four developing technologies required for successful computer mediated collaboration and design [2]. The adoption of these devices is likely to be dependent on their ease of use, as consumers now expect both hardware and software to be designed for instant use – without reading the manual [3]. As such the design of the user interface has been identified as one of six core sub-fields of computer science [4]. Because many designers lack extensive computing experience their productivity will correlate directly with the usability of their tools interface [5]. The complexity of the underlying application is generally reflected in the user interface and therefore careful design is required to allow simple access to each of the operations. Data models in engineering applications tend to be more complex and so designing the corresponding user interface is more of a challenge [6]. In addition to the complexity of the application, the form factor of the physical device and its associated input/output mechanisms must be considered. To-date much research effort has been focused on desktop based human computer interaction and graphical user interface design. As such



numerous resources exist to aid the design of desktop software. Useful websites include the ACM Special Interests Group on HCI (URL 1), The HCI Bibliography (URL 2) and The GNOME Human Interface Guidelines (URL 3). In contrast relatively few guidelines are available to aid the interface design of mobile computing devices. This new style of HCI needs more research, in particular in the areas of speech and gesture interaction and 3d visualisation [3]. Until this is complete developers will continue to use guidelines issued by large software houses, which are often specific to that companies software but do offer some general advice. Examples include Microsoft's Technical article on developing websites for Pocket PC Internet Explorer (URL 4) and Macromedia's Pocket PC authoring guidelines (URL 5). A concise overview of mobile device usability is also presented by Buchanan [7].

Data Representation

Although improving rapidly, PDAs still suffer from: 'small screen size, slow text input facilities, low bandwidth, small storage capacity, limited battery life, and slow CPU speed' [8]. Of particular importance is the screen size and resolution, for example it has been demonstrated that small screens have a detrimental effect on browsing related tasks [9]. In general the [presentation] problem can be defined as 'Too much data, too little display area' [10]. The most common solution is to scroll (in one or two directions) the data into and out of the visible area. This technique has the disadvantage of occluding the remaining data, therefore removing the context of the visible data. Remedies to this include: a separate display of context, a magnification of the focus overlaid over the context, a semi-transparent focus overlaid on the context. Each of these has their relative advantages, but all suffer from continuity problems. That is, it can be difficult to trace related data from the overview (context) into the area of focus (and vice-versa). This has been addressed using distortion based techniques [11] such as the bifocal display. The bifocal display focuses the centre of the screen on the important data and distorts the remaining data (either side) such that it is still in view but not in focus. The encoding used for this contextual information is crucial, as an overview of the information space should be immediately available. Due to the extremely limited amount of screen area these techniques may not be appropriate to PDAs. An alternative to distorting the data is to rearrange its layout. For example a long list can be transformed from one to two-dimensions by 'folding-up' columns of data side-by-side [12]. A concept concerned with 'accessing the underlying data structures of an application and reformatting the data to generate a modified view' [10]. In this condensed format it may be desirable to obtain more information about a specific item, this is achieved by allowing the item to be queried. An example of dynamically generated lists, designed to improve a web browsing on PDAs, is presented by Buyukkokten [8 & 13]. If the items of data are not easily condensed into an ordered list, for example a collection of images or Internet banner adverts, then 'rifling' techniques may be more appropriate. Here each item is displayed in succession for a short period of time such that the user can gain some idea of the contents of the list. An implementation of the rifling technique is Rapid Serial Visual Presentation (RSVP) of web pages on small displays [14]. Here each page displayed prior to the current one is stored in a list of footprints and each link from the current page is stored in a list of out-links. It is then possible to show each member of either the footprint or out-links list in rapid succession. This helps the user identify their path to the current page (and back track if necessary) and locate a suitable subsequent page to visit.

Engineering Analysis

The objective of the project is to establish whether handheld computers in conjunction with network access are useful for performing engineering analysis. Engineering analysis in the broadest sense may range from paper based calculations to super-computer based mathematical models. These represent extremes both in the level of input and output data required/generated and the number of computation cycles needed to find a solution. Handheld computers are limited on both accounts. However if the processing can be carried out remote from the device then the only remaining barrier to the mobile use of this software is its interface. The choice of engineering application is therefore restricted by its input and output requirements. At present the combination of limitations in; screen size/resolution, screen colour depth, processor speed, memory and input modes mean it is not feasible to enter complicated geometric models. Furthermore the aim of the project is to allow the analysis to be performed by the engineer whilst discussing design options with a 3rd party. As such time and resources will limit the extent of input available to perform the analysis. Based on these limitations an existing Façade analysis tool was selected for use within the project.

Façade Analysis

The Façade analysis tool, NatFac, developed by Arup Research & Development was identified as suitable for the project. NatFac, written in the FORTRAN programming language, is designed to analyse the thermal performance of ventilated façades. Its input is based on a flat file which describes:

1. The façades orientation, tilt, exposure levels and surrounding ground reflectance.
2. The façades location and local weather conditions.
3. The depth of the façade, the adjoining spaces temperature, shade factors and the number of floors.
4. One floor is defined based on a number of modules.
5. For each module: Name, Height, Inner Fabric, Presence of a blind, Outer fabric.
6. The ventilation mode, flow rates, heat transfer coefficients and analysis period.

Two of the above warrant further clarification:

- Location/Weather: The weather data is based on 9 parameters particular to a given location. NatFac presently uses a filename to access one of 66 possible data files.
- Inner & Outer Fabrics: A fabric is defined using an external program to construct the appropriate series of layers. Each fabric is then referenced using a unique identifier.

NatFac processes this information and calculates:

1. Hourly external temperature.
2. Hourly direct, diffuse, ground and total solar radiation.
3. Transmitted solar radiation for each module of the floor.
4. For each Floor for each hour of the analysis the following are calculated: Flow rate, Leaving air temperature, Solar Gain, Load on adjacent space and the internal surface temperature.
5. Mean, Light and Heavy weight CIBSE Solar Gain Factors [15] for each floor.

This data is used to assess the feasibility of different façade constructions during most stages of the design. By implementing a version which may be accessed remotely it is expected that the software will become more widely used, and more specifically used during consultation with clients and architects.

System Infrastructure

Having already established that a client-server application is required to provide sufficient processor power it was necessary to determine the optimal method for implementation. As the client device is a PDA two options were available:

1. Write a standalone interface: In the case of PocketPC devices, such as the Compaq iPAQ, this would require the use of either the C++ or VB SDK for WindowsCE. A separate interface would also be required for PalmOS devices.
2. Use a web based interface: Both PocketPC and PalmOS devices are designed to access the Internet. A web based interface would be accessible to both, but have the disadvantage of a limited set of web controls.

Historically PDA programming has proved difficult, in part because of limited programming environments and as such it is advantageous to avoid platform dependent languages and adopt widely available protocols and infrastructures [16]. Option two was therefore selected, as it has the additional benefit of being accessible from any internet capable device, for example the engineers' desktop computer. The infrastructure of the resulting web based system, called Flash1, is illustrated in Figure 1. The software used to implement Flash1 is now described followed by an overview of the hardware used to remotely access the website.

Software

Flash1 is implemented as a web based application using Microsoft's newly released Visual Studio .NET (URL 6). The framework includes Active Server Pages (ASP.NET) and Visual Basic (VB.NET), where VB.NET is used to program the logic behind the visual interface constructed in ASP.NET.

The ASP.NET website:

1. Dynamically generates a device/platform independent HTML interface.
2. Reads and writes an input/output file (based on XML)
3. Starts the analysis by interfacing with a Visual Basic wrapper (a VB6 COM object) which reads the input from the users file, starts NatFac and writes the output to the same file.

ASP.NET was selected for this project because it generates HTML forms specific to the client's browser. Note: Pocket Internet Explorer (PIE) is HTML 3.2 compliant; refer to URL 7 for PIE 2000 and URL 8 for PIE 2002. This allows the developer to focus on content generation and layout rather than browser specific design. In addition, the .NET framework is specifically designed to process XML (URL 9) documents which make it suitable for creating and editing the required input/output file. Finally nearly all processing is carried out on the server resulting in a reduced load on the client device. Whilst the reduced load is an advantage, server side applications impose two restrictions. Firstly the choice and layout of web controls are limited (by HTML 3.2) and secondly certain user interactions require a round trip to the server adding a sometimes unacceptable delay [16]. It is therefore useful to limit the number of pages on which controls interact with each other as the page must be resubmitted each time the controls value changes.

Input/Output File

XML was chosen to implement the file required to store the users input and the corresponding analytical results. This had several advantages:

1. Well structured child/parent relationships.
2. Both machine and human readable.
3. Data type and format maintained.
4. Web based protocol brings it in line with other technologies used.

XML uses a schema to define the structure of the files contents. Using an appropriate schema allows the document to be easily interpreted by the reader and navigated via software. A search for an appropriate schema identified the Green Building XML schema (gbXML). The gbXML schema was submitted to the aecXML for inclusion in June of 2000 and is the draft schema for the Building Performance & Analysis Working Group (URL 10). Whilst this schema was far too extensive for the project it was felt that the relevant sections could be utilised/simplified as necessary. As the project progressed it became evident that implementing an XML schema that reflected the various input/output screens was more beneficial than following an industry defined schema. For example an index page can be created based on the presence (or otherwise) of the 1st level of parent nodes within the document. The use of the XML file to alter the appearance and functionality of the interface is outlined below.

Hardware

The application server is an 800Mhz Intel Pentium machine operating Microsoft Windows 2000 and Internet Information Services (IIS). This machine uses the .NET framework (URL 11) to server ASP.NET web pages to the Internet. These web pages are displayed on a PDA, in this case a Compaq iPAQ H3870. The iPAQ PDA was selected because its operating system (MS Pocket PC) was familiar to both the developer and users. For this project the internet is accessed using a General Packet Radio Service (GPRS) enabled mobile phone (e.g. Ericsson T39 or T68). GPRS enabled networks offer 'always-on', higher capacity, Internet-based content and packet-based data services, for further details refer to The GPRS Applications Alliance (URL 12). The two devices were connected wirelessly using the inbuilt Bluetooth functionality. Bluetooth wireless technology is a worldwide specification for a low-cost radio solution that provides links between mobile computers, mobile phones and other portable handheld devices. The specification is developed, published and promoted by the Bluetooth Special Interest Group (SIG), see URL 13. Initial testing of the Flash1 was conducted using a Pocket PC emulator (URL 14); this was followed by tests using live network access via a Wireless LAN (URL 15) Compact Flash card.

User Interface

The overriding aim of the user interface was to make it quick and simple to enter the correct data. Four key stages to the analysis were identified; select or create a file, enter or edit input, perform the analysis and review the results; these constitute the main menu (Fig 2A). All but the first option on the menu are unavailable to the user until they have selected or created a new file. User files are stored on the server within unique directories (based on the users' login name). When the user requests the ASP.NET page for selecting files the server locates the appropriate directory and lists all existing files in descending date order. Once a file has been selected or created the user is returned to the main menu which will now allow data to be entered. Note: if the existing file has a complete set of input data then it may also be

analysed (a one click operation). Likewise if the existing file contains results then these may be reviewed. This dynamic menu is based on the presence of certain nodes within the selected XML file (Figure 2A).

Input

Due to the modes of input used with PDAs it is desirable to reduce the number of 'keystrokes' and 'manipulations' required. At the simplest level this means reducing textual input and ensuring that the required screen space is less than the available resolution (otherwise scrollbars will be required). Practically this results in a limitation of no more than four one line input controls each with two lines of associated text plus a heading and a submit button. As the project progressed it became apparent that scrolling in at least one direction would be required in order to generate a usable interface (the extreme opposite being one or two input fields and their text per page). Several techniques were used to appropriately structure the input data.

Database

To simplify user interaction with the application the weather data was placed in a database (DB) and a summary presented to the user on selection of a location. The location data is presented as an alphabetical list of site names with the latitude, longitude and altitude in brackets, see Figure 2B. Once selected the location and weather data can be accepted as defaults or edited by the user, see Figure 2C.

Tables

To reduce the time required to receive each page of the interface images were used sparingly. Instead a combination of fonts, web controls and tables were used to construct a clean and functional interface. For example, a table is used to represent one floor of the façade. All floors are identical and are constructed from a series of vertically stacked modules. Each row of the table is used to represent a summary of a module and provide links to: add a new module above itself, edit its details and delete it (Fig 2D). Here the server generated tables are used to give visual structure to data which may have otherwise required complicated images. As previously mentioned each module is defined by an inner and outer building fabric plus a central cavity. Again a table is used to represent the physical location of elements (fabrics) within the façades modules (Fig 2E).

Supporting User Actions

Each fabric consists of between 1-5 layers which were previously constructed using an external interface. To avoid lengthy input times for these fabrics a set of over fifty standard constructions [15] were imported into the database and referenced in a single web control. Due to restrictions imposed by the analytical model the total number of layers used in the modules construction should not exceed ten. Therefore each time the user selects a fabric from the web control the information is posted to the server where a check is performed on the number of layers. This is used to dynamically update the interface to inform the user of the number of layers currently in use. This is an example of how the interface can be designed to support the user in entering appropriate data for the analysis engine. Further examples are:

1. The use of default values when entering 'new' data.
2. The use of a clear input description which includes details of sensible value ranges e.g. [10 to 100].
3. The use of control validation – a check is performed on each control to determine that the value entered is of the correct type and order of magnitude.

Dynamic Interface

A final technique which can be used to ensure appropriate input data is obtained is the use of dynamic input panes. A pane is used here to group controls which are co-dependent but are only required under certain circumstances. For example if the user decides to mechanically ventilate the façade then a fixed flow rate is required else if the façade is naturally ventilated then an initial flow rate and a pressure drop are required. Presenting all of these input to the user would result in an unnecessarily complicated interface, instead they are placed in two separate dynamic panes (Fig 2F). On selecting mechanical ventilation the server displays the corresponding pane and hides the natural ventilation pane (and vice-versa). Whilst this operation could be carried out using client-side scripts it was decided that to ensure platform independence it would be handled by the server. This has the disadvantage of requiring an additional roundtrip from the client to the server and back again, thus using additional bandwidth.

Output

Having already identified the screen size and colour depth as a problem, particular attention was applied to the presentation of the analysis results. As with the input the decision was taken that scrolling the page in one direction was acceptable. This has implications for both tabular and graphical presentation techniques.

Fixed Tabular Results

Tabular results are used to present a formatted set of structured data which requires the values to be read accurately. The choice of table layout is governed by the size of the data set to be displayed. In the case of twenty-four hourly values of temperature it is possible to arrange the table in two halves (folded) such that all the data is visible without scrolling (Fig 3A). In the case of tables with two to five columns it is possible to display the data whilst only scrolling in one direction (Fig 3B). This does however require the column headings to be shortened quite considerably. The full column headings may however be made available on a separate page linked from the abbreviated versions. If longer headings are required (i.e. if they are entered by the user and therefore difficult to automatically abbreviate) then the table can be transposed and the labels placed on the rows. This technique is used to display the transmitted solar radiation results for each module of the façade (Fig 3C). Here the time series is split into two (1-12 & 13-24) to reduce the horizontal scrolling required to see all the data (Note: the number of modules within the model is likely to range from 1 to 5). As the modules are placed on top of each other this order is maintained when displaying the results.

Graphical Results

To avoid the user having to scroll in both horizontal & vertical directions the height of all graphics were limited to the vertical resolution of the PDA. The resulting graph, must be scrolled along the x-axis, typically used to represent time (Fig 3D). This image illustrates the usefulness of graphs for showing overall trends in data. Graphs are also ideal for highlighting differences between variables, for example direct and diffuse solar radiation (Fig 3E). The same technique can be applied to comparing data associate with each module (Fig 3F). Again this introduces problems with the length of the label associated with each series.

Interactive Tables

Whilst fixed tables are sufficient for a limited amount of data they would not be suitable for multidimensional data for many variables. Instead an interface which supports selection of individual variables is required. The main data produced by Flash1 consists of three dimensions: hourly values (1-24) for eight variables (Air temperature, velocity etc.) for each floor of the façade. In this case the solution is to implement a two-dimensional table which is generated by fixing the third dimension. For example, by fixing the hour it is possible to query the XML file for data related to each of the eight variables (columns) for each floor (rows), see Figure 4A. By selecting a variable the page is regenerated using this as the fixed dimension (Fig 4B), the same occurs when a floor is selected (Fig 4C).

Future Work

A working concept has been presented and each step in the process tested/documentated by the developers. The next stage is to perform a user trial on a prototype system. The prototype is to be comprised of three sets of mobile phones and PDAs. These will connect to a single Internet server running Flash1. The user trial will involve briefing several engineers on the systems use, issuing the equipment for a set period of time and performing semi-structured exit interviews. The interview will seek technical, client and personal feedback. In addition the systems URL will be published internally allowing all interested parties to gain access. The use of the system will then be monitored and if sufficient traffic is detected an online questionnaire will be posted.

Conclusion

Techniques suitable for collecting complex input through a mobile web page have been identified and implemented alongside mechanisms for representing multiple output variables on small display devices. These have been used with server side programming to demonstrate how mobile computing devices can be used to perform engineering analysis. As these devices increase in computational power (with 1Ghz machines on the horizon) it will become possible to perform the analysis calculations locally, thus eliminating one of the needs for remote network access. However the form factor of these devices is

likely to remain relatively unchanged due to the maximum size of a portable screen. Therefore further research is required into how to design effective ‘small-screen’ interfaces for complex engineering applications.

References

- 1 Rafiq, M. Y., Matthews, J. D. & Jgodzinski, P. 2001, ‘Interactive role of human-computer interaction in design’, *8th Int. Workshop of the European Group for Structural Engineering Applications of Artificial Intelligence (20th – 22nd July 2001)*, Centre for Innovative Construction Engineering, Loughborough University, UK
- 2 Mitchell, W. J. 1995, ‘The Future of the Virtual Design Studio’, in *Virtual Design Studio*, Ed J. Wojtowicz, HK Univ. press, HK.
- 3 Myers, B.A. 1993, ‘Why are Human-Computer Interfaces Difficult to Design and Implement?’, *Carnegie Mellon University School of Computer Science Technical Report no. CMU-CS-93-183*.
- 4 Hartmanis, J. 1992, ‘Computing the future: committee to assess the scope and direction of computer science and technology for the National Research Council’, *Communications of the ACM*, 35(11), pp. 30-40.
- 5 Gross, M. D., Yi-Luen Do, E., McCall, R., Citrin, W., Hammill, P., Warmack, A. & Kuczun, K. 1998, ‘Collaboration and Coordination in Architectural Design: approaches to computer mediated team work’, *Automation in Construction*, Vol. 7, pp. 465-473
- 6 Raphael, B., Bhatnagar, G. & Smith, I. F. C. 2001 ‘A model based approach for the creation of flexible graphical user interfaces’, *8th Int. Workshop of the European Group for Structural Engineering Applications of Artificial Intelligence (20th – 22nd July 2001)*, Centre for Innovative Construction Engineering, Loughborough University.
- 7 Buchanan, G., Farrant, S., Jones, J. & Thimbleby H. 2001, ‘Improving Mobile Internet Usability’, *Proceedings of the Tenth International World Wide Web Conference*, pp. 673-680.
- 8 Buyukkokten, O., Garcia-Molina, H., Paepcke, A. Winograd, T. 2000, ‘Power browser: efficient Web browsing for PDAs’, *Proceedings of the CHI 2000 conference on Human factors in computing systems*, pp. 430-437.
- 9 Mohd-Nasir, N. 1999, ‘Improving Web interaction on Small Displays’, *Proceedings of 8th International WWW Conference*, pp. 51-59.
- 10 Spence, R. 2001, *Information Visualization*, ACM Press
- 11 Leung, Y. K. & Apperley, M. D. 1994, ‘A review and taxonomy of distortion-oriented presentation techniques’, *ACM Transactions on Computer-Human Interaction*, 1(2), pp. 126-160.
- 12 Furnas, G. W. 1997, ‘Effective View Navigation’, *Proceedings of Human Factors in Computing Systems*, pp. 367-374.
- 13 Buyukkokten, O., Garcia-Molina, H. & Paepcke, A. 2001, ‘Accordion summarization for end-game browsing on PDAs and cellular phones’, *Proceedings of the 2001 SIGCHI conference on Human factors in computing systems*, pp. 213-220.
- 14 Spence, R. 2001, ‘Information Visualisation’, *A one day course presented at The International Conference on Information Visualisation*, Note: also documented in 10.
- 15 ‘Thermal Response and Plant Sizing’ 1999, in *CIBSE Guide B: Environmental Design*, CIBSE Publications.
- 16 Fox, A., Johanson, B., Hanrahan, P. & Winograd, T. 2000, ‘Integrating Information Appliances into an Interactive Workspace’, *IEEE Computer Graphics And Applications*, 20(3), pp. 54-65.

URL 1 <http://www.acm.org/sigchi/>

URL 2 <http://www.hcibib.org/>

URL 3 <http://developer.gnome.org/projects/gup/hig/hig-0.1/>

URL 4 http://www.microsoft.com/mobile/developer/technicalarticles/pie_dev.asp

URL 5 <http://www.macromedia.com/software/flashplayer/pocketpc/authoring/>

URL 6 <http://msdn.microsoft.com/vstudio/default.asp>

URL 7 <http://www.microsoft.com/mobile/developer/technicalarticles/html.doc>

URL 8 <http://www.microsoft.com/mobile/developer/technicalarticles/whatsnewpie.asp>

URL 9 <http://www.w3.org/XML/>

URL 10 <http://www.gbxml.org/>

URL 11 <http://msdn.microsoft.com/netframework/>

URL 12 <http://www.gprsworld.com/>

URL 13 <http://www.bluetooth.org/> and <http://www.bluetooth.com/>

URL 14 <http://www.microsoft.com/mobile/downloads/emvt30.asp> (Emulator in eMbedded tools)

URL 15 <http://www.wlana.com/>

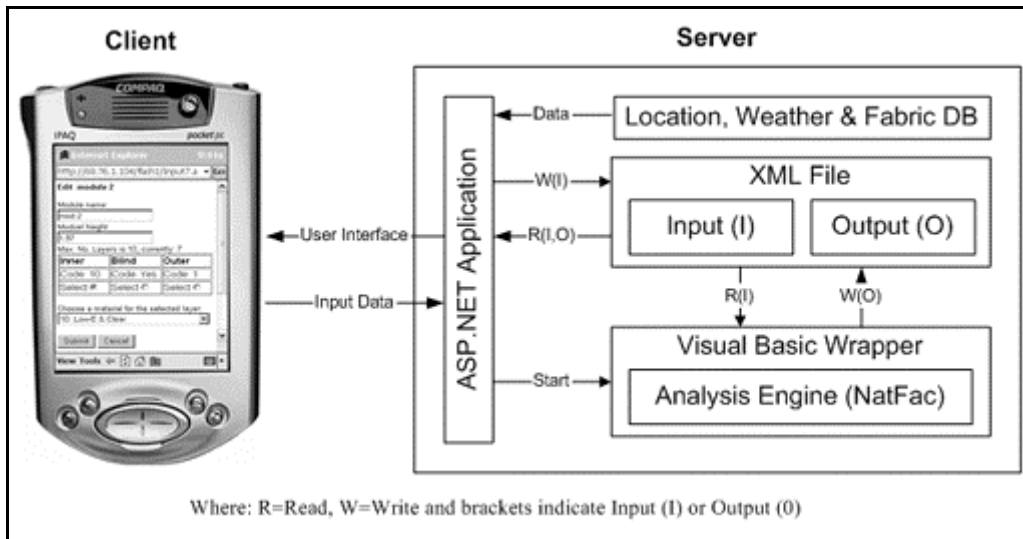


Figure 1- Flash1 Client-Server architecture.

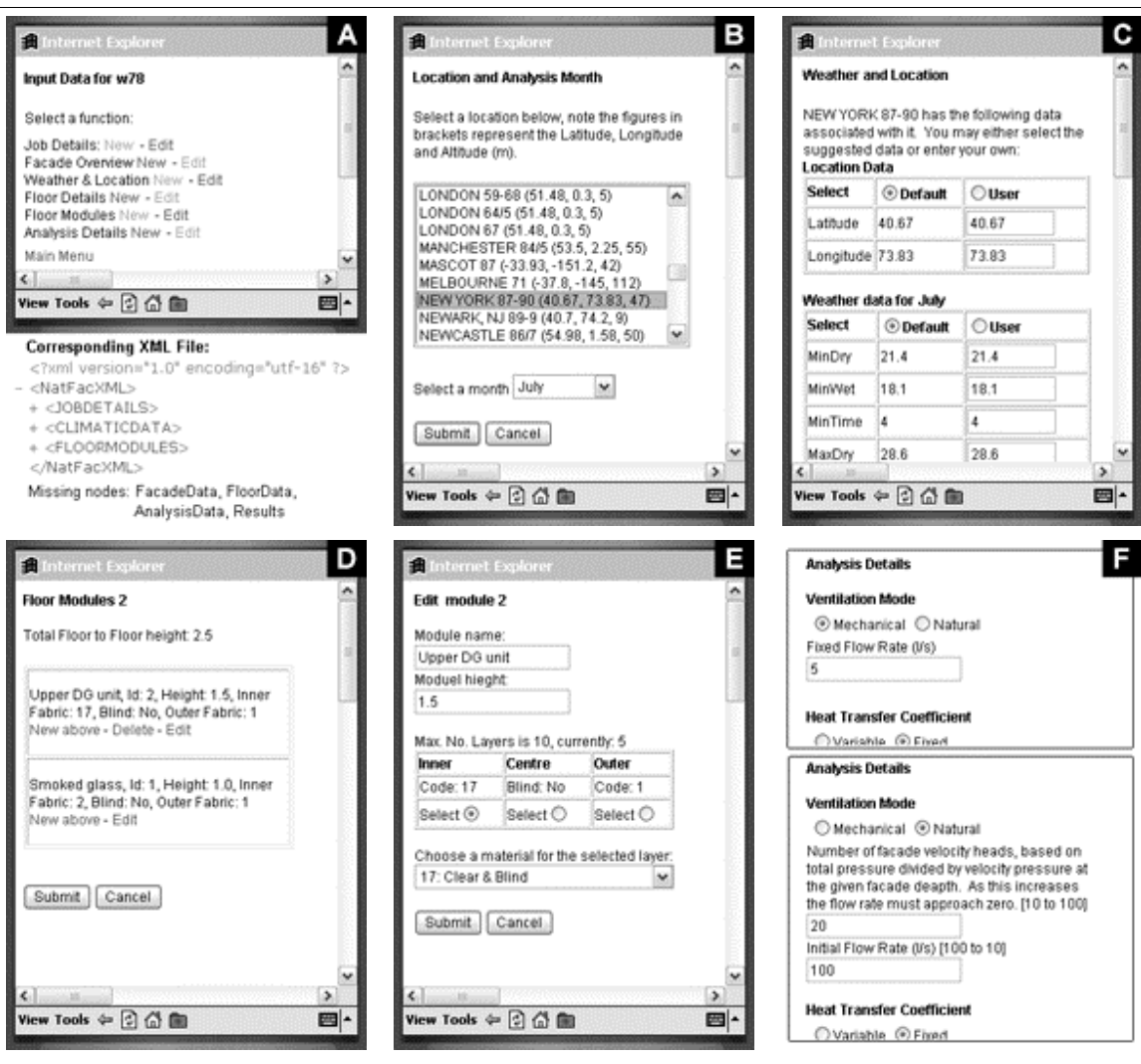


Figure 2 – Examples of the input interface.

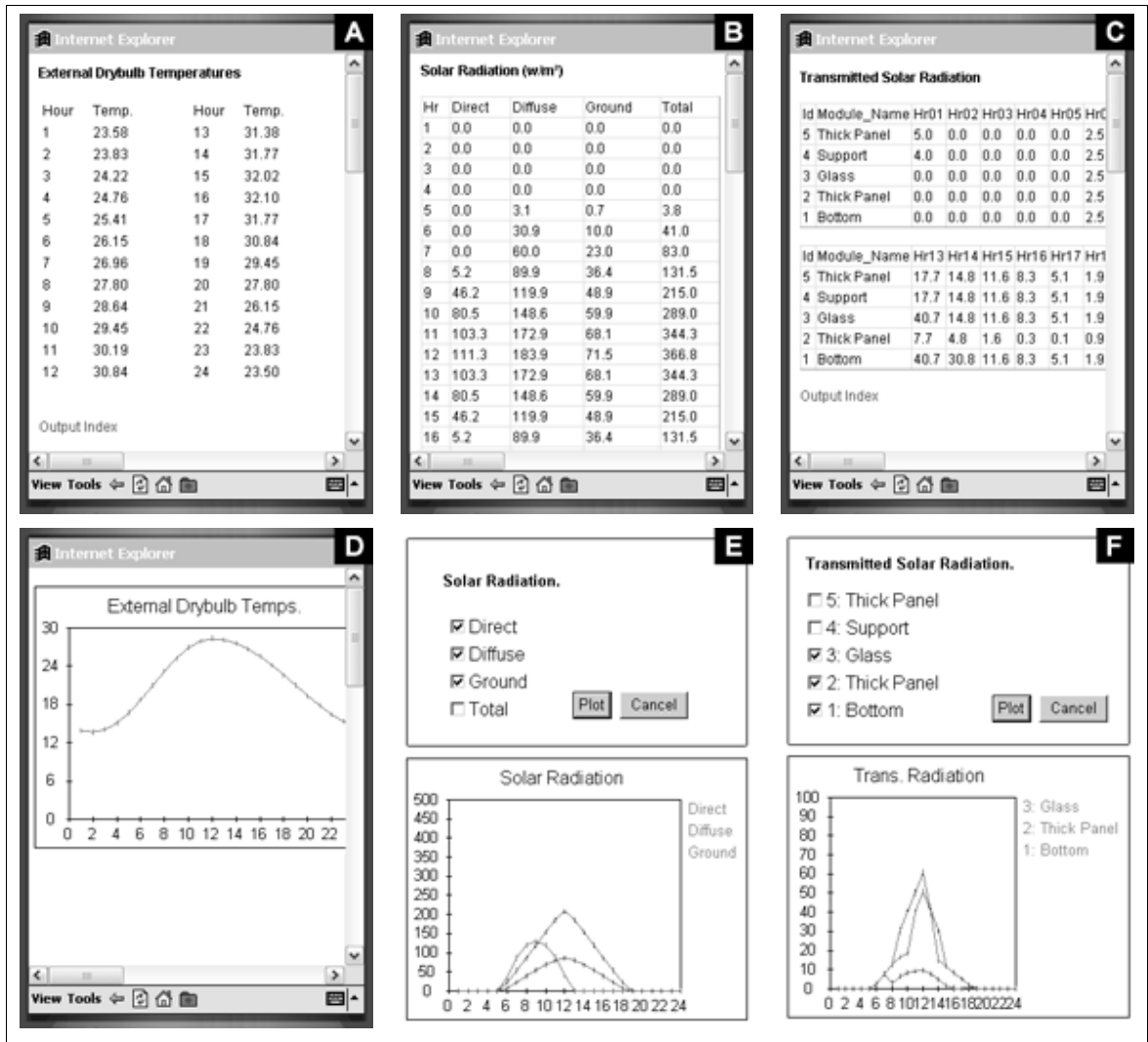


Figure 3 – Examples of static output interface.

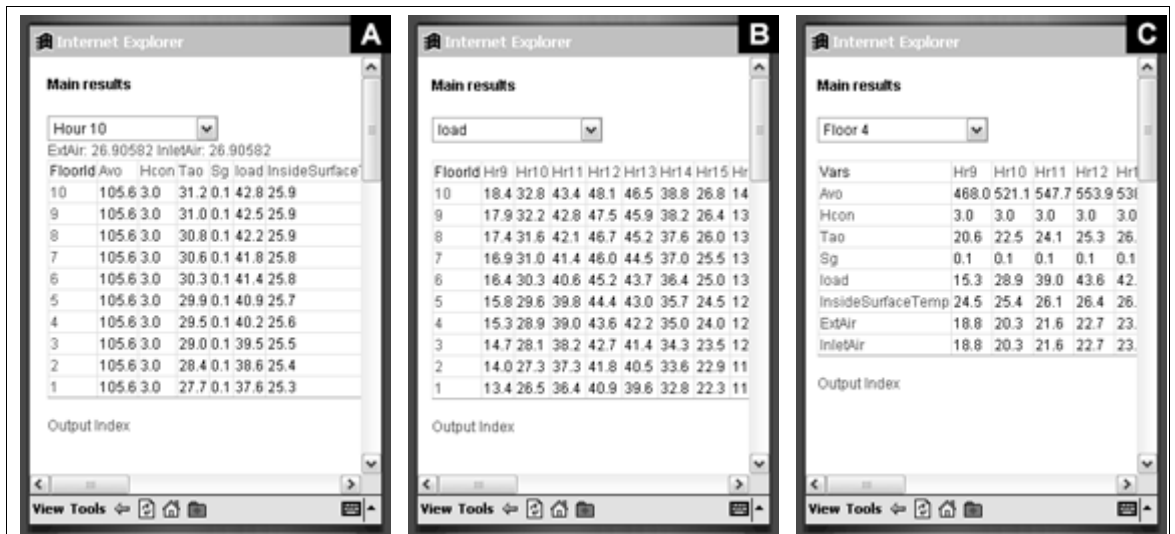


Figure 4 – Examples of dynamic output interface.