Context-aware mobile interfaces for process automation

Mikkel Holm Olsen
mho@oersted.dtu.dk

Morten Lind
mli@oersted.dtu.dk

Ørsted•DTU, Technical university of Denmark

June 15, 2006

Handheld wireless devices are gaining acceptance and more widespread usage in process automation, in particular for field operators. Limitations in input and output capabilities cause the interfaces to become cumbersome, due to the wealth of information and limited navigation capabilities. An architecture and prototype implementation for using context-aware computing and positioning to limit the amount of information is presented. The implementation has been integrated with the HAMBO boiling water reactor simulator at Institute for Energy Technology / OECD Halden Reactor Project*, Norway. Further possibilities for using such interfaces and their benefits are discussed, and it is concluded that these concepts can provide valuable additions to the information available for field operators.*

Keywords: Context-aware computing, Human-machine interfaces, Automation systems, Handheld device, PDA

1 Introduction

Through several decades the level of automation in various types of factories has been rising. More and more different processes are automated, while the level of automation of the single process is also rising, thus improving the efficiency and throughput of factories, and hence economic gain.

More and more information about the process is remotely sensed and presented on panels in the control room. Since the control room provides a much more comfortable working environment for the process operator, there is a great demand for this centralization.

However, some tasks can still not be solved from the control room. These involve maintenance and other tasks where physical presence of the operator in a particular location in the process system is required. In these situations a field operator will undertake the task at hand, in most cases with extensive communication with one or more operators in the control room. This communication is mostly verbal, in the form of telephone or two-way radios (walkie talkies).

With the centralization of information in the control room, the field operator needs to request a lot of information about the system state from the control room operator. There may also be a need to change some parameters of the control system, as well as shutting down or starting up various subsystems or components. These tasks are carried out by the control room operator, under request of the field operator.

The advent of wireless digital technology provides an ability to present the field operator with an additional interface to the control system, yielding information about the state of the system, and if desired, the ability to interact with the system as well. Although this may be viewed as a decentralization, we consider it a centralization in a location that is not physically fixed, but corresponds to the location of the field operator. At the same time, however, it should be considered a distribution of information and control, since there is no intention to remove anything from the control room.

For the interface provided to the field operator, a Personal Digital Assistant (PDA) with wireless capabilities is suggested. This device can provide a highly portable display as well as basic input capabilities, allowing the field operator to interact with the process system. Of course, other types of portable devices, such as tablet PCs and wrist-worn computers, or even mobile phones, could also be used.

One major issue with this kind of "portable control room" is the limited input and output devices on handheld computers in comparison with the panels present in the control room. The display on a PDA has very limited screen area and resolution compared to a single monitor in the control room, let alone multiple screens and projectors. It is therefore suggested to use additional information to filter the data presented on the handheld device. In essence, this will constitute a context-aware system that provides the field operator with information about the process system and the ability to interact with it.

In this paper we present a suggestion for an architecture that can be used to integrate these context-defining parameters [DAS01] with an automation system and a human-machine interface. A prototype implementation is presented showing some of the possibilities of this approach. Finally, we will discuss some of the benefits provided by this system and the possibilities for further expansion of the concepts.

2 Defining context

In order to limit the amount of information that needs to be presented on the field operators handheld device we need to find ways to determine what kinds of information is relevant. In other words, we are trying to find parameters which define the context.

The most obvious parameter defining the context is probably the state of the automation system as a whole, or of the various sub-systems. In fact this type of context information is already used to change the appearance of user interfaces in automation systems, *e.g.* functional displays, ecological displays [Vic99].

Another type of context parameter that could be very useful is the intentions of the field operator. For example the operator could let the system know that a particular maintenance task is being carried out (such as replacing a pump). In this case we can think of the information as defining some part of the "state" of the operator. If we furthermore have information about the existing procedures for carrying out the task, this can be a valuable source of information for the presented display, and could also allow the system to help ensuring the procedures are correctly followed.

The way to get most accurate information about the intentions and tasks of the field operator is to ask him. Unfortunately, this gives an additional workload for the operator. The system will also appear more cumbersome if this information needs to be input before the task can be carried out.

A suggestion for assessing the intentions of the operator is to use a positioning system to determine the location of the field operator in the automation system (in reference to the components of the automation system). While this information is not as accurate as knowing the exact intentions and tasks of the operator, its main benefit is that it does not require the operator to provide any additional information. Various technologies for determining position (in this case of the field operator) exist, and simply mapping the acquired position to a list of control objects of relevance seems fairly easy, but yet appears to give good results when trying to determine what objects may be of relevance to the task being carried out. In this study it has therefore been decided to examine this subject in greater detail.

The position is mainly related to the intentions of the operator, so this information can also be used in conjunction with procedures for various tasks. Even if the operator is requested to input the task being carried out to the system, we can still benefit from also having the position, since this may help determining the intention of the operator in relation to some sub-tasks etc.

Time seems like an important parameter when defining context. In some sense it could even be considered fundamental, since all information about the process is logged with reference to time.

In most cases, however, the actual wall-clock time is of less importance. The actual point of interest is often related to the state of the automation system, but in some situations this state information is not internally represented in the system.

When we consider creating a logic function to assess some state information based on time, we are probably trying to find some recurring event, as opposed to something that happens only once. If the event occurs only once, it is most likely easier just to wait until the event occurs and manually take action, as opposed to programming behavior that is only used once.

Such recurring events could be related to some cycle of the system, such as a production batch. There could also be other cycles that occur at different frequencies.

A case where the wall-clock time seems important could be a production facility that runs around the clock. The operators have different shifts, *e.g.* 8:00–16:00, 16:00–24:00 and 24:00–8:00. In this case it might be sensible to make the system or interface aware of these events, so that the next team of operators can be properly briefed about any abnormalities that have occurred during the past shift. We can see that the system is assessing some state of the environment (namely the change of shifts) based on the time of day. Similarly, the production requirements of some product may be related to the season of the year, or a particular part or subsystem must undergo maintenance in the beginning of each month. These types of information could be useful when incorporated into the system.

A very common scenario where time is important is when a time span or interval is used. For instance a heating element should be turned on for a certain amount of time. Notice that this is (normally) not related to the wall-clock time, but to the time span between two events. We may say that some event should follow a predetermined amount of time after some other event, *e.g.* the heating element should be turned off 18 minutes after it was turned on.

An automation system could also benefit from having information about the state of the environment, such as the current demand for something being produced. In some domains, such as electric power production, this is particularly important. Since there is no buffer, the production has to match the demand in real-time. In other domains, the issue is much less critical, because any stock of the product acts as a buffer of the plant output, which will handle changes in the demand and allow the adjustment of the production to "follow up".

Although the demand for electricity in a power plant is imposed externally (by the consumers), we can in some sense view this as an internal parameter, since it can be measured by the system in real-time. In cases where a parameter is based on some state of the environment, the "interface" to the system is normally an operator. In these cases the parameters can be thought of as a state information about the operators intentions, since the operators goals are based on these parameters of the environment.

3 Architecture

Having discussed the parameters that define the context for an automation system, we

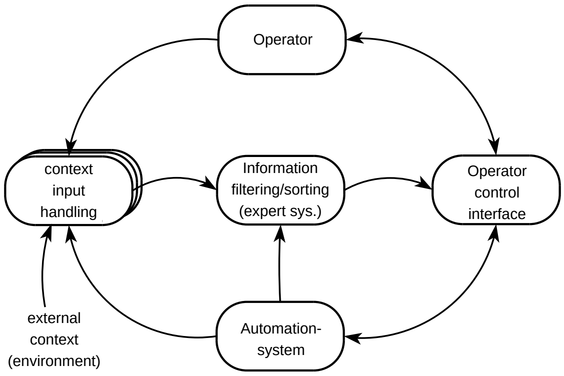

Figure 1: Overall architecture

will now turn our attention to the architecture suggested for handling these context-parameters and merging them into something useful for the presented interface.

The proposed general system architecture is depicted in figure 1. One or more context input handling modules assert various states of the context, both of the operator, the automation system, and the environment.

The filtering and sorting is handled by a separate software module, which controls the appearance of the display on the operators handheld device.

In cases where several types of context information need to be combined, it is suggested to use a knowledge based system(*e.g. an expert system*), because deciding what information to present on the operator display will most likely be highly heuristic. The main advantage of knowledge based systems over a traditional software module written in a procedural language is the ability to create complex behavior by writing relatively simple rules, and it can be a very efficient way to solve problems where no general "algorithm" exists.

The rule base of the system allows specific knowledge of the process and automation system to be specified as rules and guidelines, which is often the way a process operator may think of this kind of information.

The use of a knowledge based system also allows the separation of the rules and logic from the user interfaces, positioning system and other external modules providing context information. Integrating an extra type of context information can be achieved by simply adding rules to the rule-base.

4 Case study

In order to get some experimental results, it was decided to develop a prototype setup with the main focus on using the field operators position to assess the intentions of the operator.

The case study was carried out at *Institute for Energy Technology*[1], Halden, Norway (IFE). The automation system is the HAMBO [PNN01] boiling water reactor simulator, one of the nuclear simulators in IFE's HAMMLAB2. HAMBO is used for research and experiments on man-technology-organization issues. The control room operators' interface for HAMBO is implemented using IFE's graphical user interface management system, ProcSee[2] (previously Picasso[JS02]), and it was decided to use ProcSee to implement the field operator's display of the prototype as well.

In the prototype, the positioning system is simulated by a virtual reality (VR) model and a VR-viewer modified to output the location (and orientation) of the viewpoint. This approach eliminates problems with accuracy or limited coverage, that may exist when using a real-world positioning system. The architecture allows for other (real-world) positioning systems to be used as plug-in replacements for the VR-based module. The VR model of the plant had previously been created for a HAMMLAB experiment denoted *Extended Teamwork*[SSJ05], and was used without modifications.

Since the prototype is only concerned with the operators position for defining the context, it was decided that there was no need for a knowledge based system for the information filtering/sorting module, so this module was made as a simple C-program, which also allowed the use of the ProcSee API to interface with the process system and user interface.

[1] `http://www.ife.no`
[2] `http://www.ife.no/procsee/`

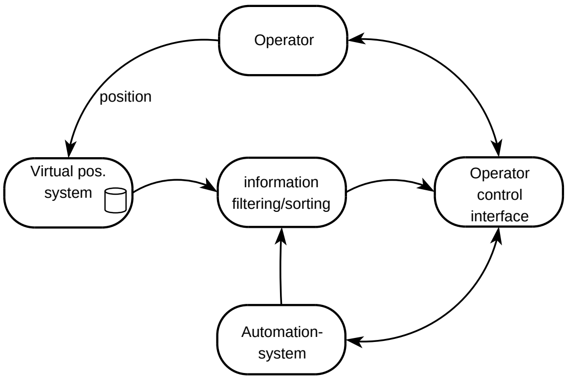

Figure 2: Simplified architecture

The simplified architecture is depicted in figure 2. As shown on the figure, the virtual positioning system has a local database. This database contains static positions of the objects in the (virtual) automation system, because this information was not provided by the VR-viewer.

The object positions in the database are used in 3D geometry calculations in combination with the operator position (and orientation if available), maintaining a list of the objects distances to the operator. This list is passed on to the information filtering and sorting module, which in turn controls the appearance of the operators interface.

The orientation of the operator is used to calculate a view-cone (determining what is in the operators field-of-view). In a similar fashion, we consider each object to have an "operation-cone" inside which it is possible to operate the object. This is depicted in figure 3. For instance, if a valve is mounted on a wall, and the operator is located on the opposite side of the wall, it will not be possible to operate the valve, even though the operator is close to the object. The threshold angle for the operation-cone and the object orientation can be different for each object, since these are specified in the local database.

In order to organize the information displayed on the operators interface, it was decided to use a design consisting of a list of objects. The types of objects is of course highly domain specific, but in the prototype we have chosen components such as valves, pumps, tanks and heat-exchangers.

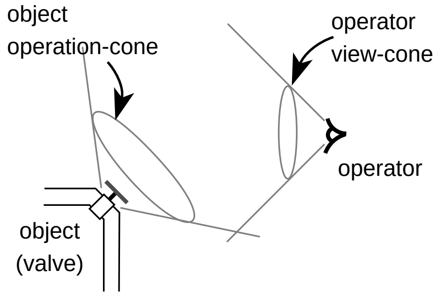

Figure 3: Operation- and view-cones

The information filtering and sorting module filters the objects based on the view-cone (if the operator orientation is available) and the operation-cone, and the list is sorted and presented to the operator.

At the time when the case-study was carried out, it was not possible to run ProcSee on a PDA, so it was decided to emulate the limited screen size of the PDA simply by creating the user interface in a small window on a desktop computer. Of course this approach has some drawbacks when evaluating the user interface in relation to the constraints on input capabilities, but it was deemed sufficient to prove the concept of such interfaces.

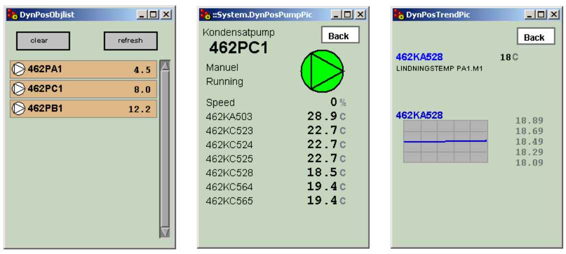

Figure 4: Example screen-shots

Some of the screens (pictures) of the implemented prototype are depicted in figure 4. The leftmost screen shows the list of objects, in this case only showing three pumps. The listed objects can change to reflect the current status and distance, but elements in the list will not swap positions or appear/disappear unless the operator requests an update (by pressing the "refresh" button). This strategy was decided in order to avoid problems with sudden changes to the list of objects.

The screen in the middle of figure 4 shows an example of the data presented when a pump has been selected from the object list.

The last screen on the figure shows the trend of a data value. This can be useful to the operator when trying to diagnose a problem with the object (such as the pump). While the screen does not currently display information regarding any alarms for high or low values, this would be a natural addition.

5 Discussion

The developed prototype system we have just described has allowed evaluation of the concept of using context-aware human-machine interfaces on mobile devices. More advanced methods for combining context information exists [DKL+02], but even with the fairly simple implementation of the filtering system in this study, it has become evident that using the operators location in the process system provides a good way of limiting the amount of data for presentation on a mobile device in a complex automation system.

As previously discussed, adding other types of context information can provide even greater benefits in terms of filtering the displayed objects. Some of this information may already be available in the automation system, while other types of information must be added to obtain the greatest opportunities for a context-aware system.

So far we have mostly been concerned with the presentation of data related to the process of the automation system. Other types of data could be integrated into the system in order to make it even more useful. For example the ability to display specific procedures for maintenance operations on a particular system component.

Other additional information that could be helpful is the maintenance and service history of each component, along with information concerning the expected lifetime etc.

The last two examples are both concerned with information that may not be integrated with the automation system, but since service history and maintenance procedures are assumed to exist in some form, it is just a matter of converting this information into a format that can be integrated with the system. It is anticipated that making this kind of information available on a handheld device could be very helpful to the field operator.

So far this paper has primarily discussed the benefits of a mobile interface providing information and control to a single field operator, but we can give a few examples where the system can be beneficial in collaboration and teamwork. For example the system allows the field operator to verify that a certain task has been completed by the operator in the control room. If for instance the field operator is doing maintenance on a particular system component, there will most likely be other subsystems that need to be switched off during the work. Under normal circumstances the field operator can contact the control room, where another operator can hopefully provide assistance in shutting down the needed subsystems. Once the operator in the control room confirms the subsystems are offline, the field operator can start the maintenance task. While we do not propose that the communication between the control room and the field operator should be changed, it is a fact that human communication is misunderstood from time to time (in particular when speaking in noisy environments), and thus the system suggested in this report can provide the field operator with some means for verifying that the subsystems are in fact offline.

The coupling with the positioning system can provide means of detecting another kind of errors that may occur: Performing maintenance on the wrong system component. With several identically looking subsystems, the field operator can make the mistake of performing maintenance on a different object than was intended. In some cases, such errors can be catastrophic, and any system that can alert the field operator in such situations could be helpful in providing a more safe working environment.

It has not yet been thoroughly discussed what kind of interaction with the process system is possible from the user interface of the portable device. The cases covered so far would provide benefits to the field operator by presenting information that would otherwise not be available. Of course the integration with the process system also allows the user interface to be designed to interact with the process control system. In many cases it may be beneficial to have some kind of access control which allows the field operator to only affect a small subset of the process system. The system could be coupled with a display in the control room showing which actions have been taken by the field operator.

6 Conclusions

On the basis of a discussion about the relevant parameters defining context in automation systems, an architecture for integrating context-aware computing in complex automation systems has been proposed and discussed.

Through a case-study a prototype has been implemented, which has allowed experiments with this technology for providing field operators with useful information, yet filtered to avoid information overload.

A range of options for further expansion of the system and applicability of the technology has been discussed, and it is anticipated that field operators in complex automation systems in many different domains can gain great benefits from this technology.

Acknowledgments

The case-study was carried out at *Institute for Energy Technology*, Halden, Norway. The most graceful thanks to everyone at IFE who made the stay enjoyable and provided guidance and help during the project development.

References

[DAS01] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2):97–166, 2001.

[DKL+02] N. Desai, K. Kaowthumrong, J. Lebsack, N. Shah, and R. Hano. Automated selection of remote control user interfaces in pervasive smart spaces. In *Proc. Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings*. Ubicomp, September 2002.

[JS02] Håkon Jokstad and Carl-Victor Sundling. Picasso-3 user interface management system. September 2002.

[PNN01] Eija Karita Puska, Sixten Norrman, and Christer Nihlwing. Three-dimensional core models in research simulators. In *Proceedings of ANS International Meeting on Mathematical Methods for Nuclear Applications*, September 2001.

[SSJ05] A.B.M. Skjerve, S. Strand, and G. Skraaning Jr. Extended teamwork in nuclear power plants: plans for an exploratory study. 2005.

[Vic99] Kim J. Vicente. *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1999.