Enhancing Operator Performance in Critical Real-time systems using an Intelligent, Adaptive, Agent-based system

Khalil C.J & Alty J. L
Impact Research Group, Loughborough University, UK
C.J.Khalil@lboro.ac.uk & J.L.Alty@lboro.ac.uk

Abstract

As the complexity of process plants increases, the task of the operators becomes more complex. This trend is likely to continue, and the current approach of providing unintelligent interfaces which cannot adapt to changing plant and operator conditions has to be addressed. One project attacking this problem is the AMEBICA project. Using a collection of intelligent agents an adaptive interface is proposed. The conditions triggering adaptation and likely responses are analysed and presented as an Adaptability Matrix. The Agent architecture is described and some examples of reasoning processes given.

1. Introduction

As more automation has taken place in the process control area, the role of the operator has changed from being concerned with maximising the mean time between failures to maximising the time between incidents. Automatic Supervisory and Control Systems have increasingly replaced many minute-by-minute monitoring operations, freeing the operators to concentrate upon higher level tasks. However, because of such automation, process plants are now required to operate within much tighter margins, and the economic costs of failure are much higher than previously. In a modern plant, therefore, having the right information presented in the right form at the right time is paramount from safety, economic and control, points of view. Operators need better organised, up-to-date information, which they can understand. Knowledge Based Systems have been employed to assisted in improving the quality of the information provided whilst it is hoped that well-designed multimedia interfaces will improve the understandability and relevance of the information provided.

The Multimedia design issue is not primarily a technical one. The issue is one of choosing an optimal set of media to achieve a particular goal in the right context. In process control interfaces such goals will either be:

perform the task more quickly
perform the task with fewer errors
make the task easier
make learning and remembering the task easier

All these goals influence the design of the interface, and the context will determine which goal dominates. What is clear is that no one interface is likely to satisfy all these goals simultaneously so that there is a need for a flexible interface which can adjust automatically to changing system conditions and operator requirements.

This paper describes such a flexible interface, developed within the AMEBICA (Auto-Adaptive Environment Based on Intelligent Collaborating Agents) project. AMEBICA utilises a set of co-operating agents to make decisions about the form, location, modality and magnitude of representations at the Interface. This system is currently being prototyped for two real time process control domains, a Thermal Power Plant (TPP) and an Electricity Network Management (ENM). It is hoped that, by using a flexible representation mapping system and an intelligent spatial reasoning manager, AMEBICA will aid in overcoming the information overload problem that occurs within the domains during serious disturbances.

2. Industrial Problem

The extensive use of computer technology in real-time, critical Process Control situations has put new demands on the quality and effectiveness of the operator interface required. For example, the ENM centre oversees a wide geographical area with a large number of substations (often presented as several hundred values of a scrolling background larger than the display size). When a critical situation occurs within the network, the operator has to deal with a large amount of information is a short time during the disturbance. Since the interface acts as the only communications medium between the process and the operator, the performance of the entire system is conditioned by the quality of this interface.

This situation has not been aided by the fact that recent increases in computer power have led to more powerful control systems being developed which are able to process and display much greater amounts of information. Since human cognitive processing capabilities have not risen in the same period, a gap has developed between the real time process control system output capability and the operators' ability to utilise it effectively. The gap manifests itself particularly at critical times, when the operators are overloaded with information, and still have to make vital decisions. The gap can be reduced by one of two approaches, either *improving* human ability by increased operator training *or* by reducing the complexity of the task. Ideally the two should be used in conjunction, with the system intelligently providing salient, context-based information, in such a way as to reduce the complexity of the tasks for the operators.

The problem not only lies with the amount of information presented, but also how the information actually rendered. In times of high information flow, the system will typically display relevant information in compressed, narrow time-windows utilising a variety of visual and auditory forms. Most of the presentation modalities are fixed during the interface design and system configuration period, and cannot change during plant operation. This rigidity of the interface often limits the effectiveness of information presentation, since perfectly suitable representations for normal operating conditions, are inappropriate in critical situations. Additionally, control systems typically respond in the same way, independent of the fact that the flow of information is low or extremely high, or that the level of expertise of the operator is high or low.

Additionally, current systems do not cope well when there is a sudden transition of operator workload (low when monitoring the system, high when intervening to diagnose alarms) and this may be a contributing factor to errors made during fault diagnosis [9]. If we can constrain the data supplied to the operator in some way, the apparent complexity of the system may be reduced [5,6]. What we therefore need is a dynamically adaptive system that responds to system and operator conditions.

As a result, current systems have the following drawbacks:

1. The information is presented in a rigid structure.
2. The interfaces are not designed for emergency situations where the information rate is too high and operators have to face new situations.
3. The information is usually not prioritised effectively.
4. There are navigation difficulties in current complex interfaces.

One way of solving these problems is to introduce an element of intelligent, adaptive behaviour in the presentation system. An adaptive system could offer an attractive solution to these problems, by monitoring system and operator context, reducing information overload by enhanced presentation, and by filtering data that is irrelevant to the current problem context.

3. Adaptive Systems.

Adaptive Multimedia Interfaces is a recent research area and now very active. Approaches differ through the ways and means used to obtain *adaptation* (and even in their definition of what adaptation actually is). Relational grammars are used in [6] to generate text and picture documents. Many approaches have adopted a Knowledge-Base Systems approach. Examples include an automated multimedia authoring and multimedia presentation tools approach [4][2], the development of intelligent interfaces for process control for the nuclear power industry [3] and the PROMISE project [1], which developed a dynamic choice of media to the operators at runtime.

All the above works are centred on classical Artificial Intelligence techniques. The techniques are powerful when action must be decided upon and taken in the context of a consistent and monotonic world. However in many process control situations (like our proposed industrial applications) operator awareness, system state and context are continuously changing and are sometimes in contradiction with each other. Any multimedia interface needs to achieve a balance between these three components and be flexible enough to adapt to changes in the relative importance of each of these three components. In such an environment, we have to consider adaptation as driven by the process and the operators' state. A mixed-initiative user interface approach [7] should be most favourable, where "from the perspective of decision theory, decisions about action versus inaction should be directed by *expected utility*" and "autonomous actions should be taken when an agent believes that they will have greater expected value than inaction for the user". Having investigated a number of alternatives, we considered that such an interface can best be designed and implemented using techniques that have been developed in Multi-Agent Systems.

A good model for adaptive systems is human-human communication. The participants adapt their behaviour according to the characteristics of their communication partner. Each participant has an internal model of how they expect the partner to behave, and can compare this with how the partner actually behaves to change their own behaviour as necessary. Additionally, the communication process itself enriches and refines the knowledge of both partners about each other [10]. In the case of human-computer interaction, we can attempt to define models of expected operator behaviour, and use these to adapt the interface as necessary. However, this is not enough, as the system must also have a model of what is going on in the process, so that a full context of the system is understood. This enables the adaptive system to display only the required data needed for the operator to deal with a system disturbance. Any viable system, therefore, must implement both a model of the operator and a model of the process, so that it can respond to both system and operator performance.

The general requirements for an adaptive system have been stated by Opperman [10]. These requirements include:

- deciding the most important goal to achieve. Since there are usually not enough interface resources to achieve all the goals, some of them must have higher priority, others must wait.
- paying attention to the current context and using it to control the behaviour of the interface.
- reacting to unanticipated events. This is very important when trying to reducing operator overload in emergency situations.
- interrupting actions. When the system state changes, it may be important to be able to interrupt on-going actions in order to perform other actions that have increased in importance. Interrupted actions should be recorded and perhaps can be resumed later.

The overall goal of these requirements is to reduce the need for operator request for help, anticipate the operators' needs and reduce the frequency and magnitude of operator errors [11].

4. Forms of Adaptation in AMEBICA

The AMEBICA system provides adaptation on two main fronts

- on-line selection of a rendering from a set of possible representations together with the parameters of that representation (we call this a flexible mapping)
- spatial adaptation of presentation, where the system layout manager attempts to make the display as organised and clear as possible (we call this spatial adaptation).

Our on-line selection approach differs from traditional (rigid) interfaces in that interface mappings are chosen at run-time from a set of defined mappings. This is in contrast to many current approaches where mapping decisions are made at *design time* and then fixed (usually with a one-to-one correspondence). These rigid mappings are always a compromise. Although our approach still requires all the design work to be completed at design time, we retain many alternative mappings from which the best selection can hopefully be made at run-time taking into account the current context. This flexible mapping approach involves run-time reasoning between a set of predefined alternatives.

The second form of adaptation uses run-time reasoning to create new instances of adaptation. The domain here is spatial control of presentation. It is generally accepted that as the complexity of an application grows, the operator spends a significant part of their time arranging and re-sizing windows on the screen to suit the current task/context, in addition to acting upon the information present in the windows. This principal can be illustrated by comparing users performance in windowed and non-windowed systems Bury et al. [4] found that in many cases, task completion times in windowed systems were often longer than in non-windowed systems due to the time spent in window arrangement. They found that although multiple windows reduce the user's short term memory load, they often imposed an additional management workload on the user. Our system attempts to alleviate this problem by continuously reasoning about the spatial layout and making on-line adjustments in order to maintain clarity.

It is also important that during such spatial adaptation processes, information should be moved as little as possible. Thus, we have bounded adaptation so that we usually only allow adaptation before information is placed on the screen. Generally speaking, once upon the screen, information should not be subject to large displacements unless absolutely necessary. AMEBICA will also place relevant information as near as it can to other information of the same type, and will attempt to select the best representation for that information in the current context (using flexible mapping).

When pop up windows are needed they are overlaid on top of the main Network diagram which is on the background layer (and zooms to the appropriate area), the framework intelligently decides the best placement for these pop-ups to minimise obscuring other required information, yet adhering as closely as possible to expected locations.

5. Adaptation Response

Process adaptation. New interface elements are generated in order to provide the operator with relevant information about the process without having to navigate through hundreds of subsystems diagrams and measurements windows.

Prioritisation. One of the most important functionality of the TPP application is the information discrimination. The application must present the operator with only the most relevant and urgent information. The rest of the information should be provided a few moments later.

Spatial adaptation. The interface elements previously generated (by the user or by AMEBICA) are modified in order to help the operator to see particular information: this can be done by changing the position or size of a window, highlighting a particular element in a window, etc.

Environment adaptation. Information is generated using a different medium: sound message, etc.

Measurement adaptation. The representation of an instrument (slider, trend or data) of a measurement is changed to help the operator understand what is happening. For example, when there is an error in a sensor because it does not work properly, an similar equivalent variable might be shown.

Temporal adaptation. A replay of operator actions is generated, for example, the most recent actions, which have happened in the process when the operator opens a diagram.

6. Adaptation Initiation and Response

There must be a set of conditions for starting the adaptation – specifically a set of initiating conditions or a triggering event. Secondly, there must be one or more appropriate responses or functions that can be used to respond to the disturbance. The initiating conditions mean that the adaptation functions are not activated all the time, but only when certain conditions are reached or a certain threshold has been passed. (Conversely, the functions cease when another set of conditions has been fulfilled.) Any adaptive system must therefore entail a definition of the initiating conditions and a specification of the functions that are activated when the initiating conditions are met. More generally, the initiating conditions can be seen as defining a specific goal, namely that the disturbance or deviation has been neutralised or counterbalanced. The compensating or adaptive functions must be capable of achieving this goal – without at the same time introducing new disturbances in the system or the environment.

Conditions Initiating Adaptation

The two major sets of initiating conditions for adaptation will be the state of the *process*, and the state of the *operators.*

Initiating Process States: In the case of the state of the process, it is clearly possible to identify a set of distinct system states and to associate these with specific patterns of measurements. A classical example is the state-space diagrams that can be applied to most industrial processes, comprising transitions between states such as shut-down, stand-by, normal operation, disturbance, and accident. The minimum is a distinction between two states, which can be called normal and disturbed. Since disturbances are characterised by specific indications, such as alarms, it should not be too difficult to define unequivocal signatures for each of these states. A further differentiation can, of course, be introduced if it is necessary and feasible.

In terms of process states, it is suggested to have three categories, as follows.

Normal process state: Here the process is in a normal state, as defined by key process parameters (e.g. critical functions or safety functions).

Disturbed process with high information rate: In both the second and the third category, the process is in a disturbed state. The disturbed state can again be defined based on the specifications of the normal process conditions. The difference between the two categories is whether the information rate is high or low. The information rate can be measured in several ways, but generally refers to the amount of information (signals, messages, alarms) per unit of time. Typically, the information rate is high when a disturbance occurs and for a limited period of time thereafter, but low later in the development of the disturbance.

Disturbed process with low information rate: This case corresponds to the later stage of a disturbance, when the rate of information has gone down, but the process still has not been recovered.

Initiating Operator States: In the case of the state of the operator, the situation is a little more difficult. While it makes sense to refer to specific states of the user, such as being attentive or inattentive, it is very difficult to find on-line measurements that can be used to identify these states (excluding the possibility of physiological measurements). For AMEBICA, the only available measures are those that can be derived directly from the interaction between the operator and the process, i.e., interactions via the graphical user interface and input devices (keyboard, mouse). Additional, nice-to-have measurements, would relate to external communication activities (telephone, intercom), within-team communication (comments, requests from other team members or superiors), and use of supporting facilities, such as procedures.

A main issue to be resolved is which classification of operator states should be used. As mentioned above, one candidate is *attentive-inattentive*. Others could be *stressed versus relaxed*, or *aware* (of the state of the process) *versus unaware*. These categories are neither independent, and they all suffer from being difficult to associate with specific measures, particularly given the limited range of measures that is possible for AMEBICA. An alternative is to consider the concept of the level or degree of control that an operator has over the process. While this may initially appear to be even more vague and abstract than either of the above, it turns out in practice to be possible to find at least a reasonable approximation than can be related to specific measures.

The crucial observation is that a degraded control of the system can be associated with clear performance characteristics. In general, the performance becomes more erratic as control is degraded. This means that as operators lose control there is likely to be an increase in the number of failures, delays, or incorrectly performed actions, as well as changes in the overall strategy .

The individual performance failures may manifest themselves in the number of delayed responses and in the number of incorrect actions.

In terms of operator responses, or operator status, it is proposed to make a distinction between the following four categories.

Normal response: In this case the responses of the operator are normal, i.e., the operator is fully capable of handling the situation. None of the defined indicators of loss of controlled are recognised.

Delayed response: In this case the responses of the operator are delayed. It may be possible to define the required responses and the allowed time window

Operator response	Process status normal	Process stat… disturbed, hig… information ra…
Normal	OK, no action	OK, no action
Delayed (relative to expected responses)	(1) Inattentive: **Accentuate presentation**	(4) Overloade… **Filter informa… simplify presentation**
Erratic (occasionally wrong display or commands)	(2) Inattentive. **Accentuate presentation (specific)**	(5) Overloade… **Simplify displa… remove inform…**
Disorganised (constantly wrong display or commends)	(3) Confused, loss of control. **Go to overview presentation**	(6) Severe los… control. **"Voice of god"**

Table 1 The Adaptability Matrix

of response for certain categories of events. In the case that a response is delayed, this may be used as an initiating condition for AMEBICA

Erratic response: In this case the operator fails to perform actions correctly every now and then – enough to warrant attention but not enough to be considered disorganised. It is necessary to define more specific rules for when the responses are considered erratic, and when disorganised. These rules must refer to the specifics of the domain and the application. As mentioned above, a delay may be considered a kind of erratic response.

Disorganised response: In this case the frequency of erratic responses is so high that the performance is considered disorganised. In this situation the operator has clearly lost control of the process, and his performance is no longer able to maintain the overall goals.

Adaptation Responses

The second main issue is the set of adaptive functions. When considering these it is essential to keep in mind the overall goal of adaptation in AMEBICA - to maintain a given level of performance of the joint system. The compensating functions must therefore serve to further this overall goal. This can efficiently be achieved by defining a set of more specific goals that correspond to identifiable joint system conditions. For instance, if the operators seem to be losing control and if the rate of information is very high, then an appropriate response would be to reduce the rate of information presentation, for instance by removing low priority information, filtering on categories, etc. As another example, if the operator is slow in responding even though the state of the process is normal, a reasonable compensating function would be to amplify the annunciation of the plant conditions that require a response.

Other possible adaptive responses will be altering the level of abstraction of the display, switching to a different modality or simply repeating the information with more emphasis.

7. The Adaptability Matrix

The Initiating conditions and adaptive responses can be brought together in an Adaptability Matrix. The dimensions of the matrix consist of the identifiable operator states on the one hand and the identifiable process states on the other. We currently have four operator states (*Normal, Delayed, Erratic and Disorganised*) and three process states (*Normal, Disturbed with High Information Rate, and Disturbed with Low Information Rate*), leading to a matrix with twelve cells.

The matrix that results from these categories is shown in Table 1 above. The top row and left column together show the categories of the two dimensions, while the remaining part of the matrix shows the twelve states that are the outcome of the classifications.

For each cell two descriptions are given. The first represents the state of the operator. The second description represents the corresponding goals of AMEBICA, i.e., hence the purpose and direction of the specific compensating functions that should be activated. Again, the actual contents of Table 1 are only intended to give an indication of the range of compensating functions that are required. Some of them are clearly also of an illustrative nature, such as "Voice of god" – meaning that it is necessary in some way to call external help in order to restore the situation.

One important feature of the matrix is that there are a number of cells where no response is required. This corresponds to the fact that in many situations the process runs normally and the operator is in full control. In such situations there is clearly no need to activate the adaptation inherent in the system, since the normal human-machine interface must be assumed to be sufficient. The normal situation is not confined to the normal state of the process, but may also include disturbed states where the operators have no problems in responding correctly and bringing the process back on the track. Here adaptation should be avoided, since the operators know what they are doing and need not be subjected to the potential disturbance of a changing interface.

8. The AMEBICA Family of Agents

The AMEBICA approach uses the agent paradigm and adheres to the weak notion of agency [15]**.** The total system intelligence results from the collective negotiation and communication capacities of its individual agents. This allows us to consider information from several different sources in the reasoning process. These sources will include the current environment (lighting levels, operator position etc), the state of the operator team (we use general characteristic of operators rather than individual operators), the state of the process itself, the state of the presentation interface and human factors presentation knowledge.

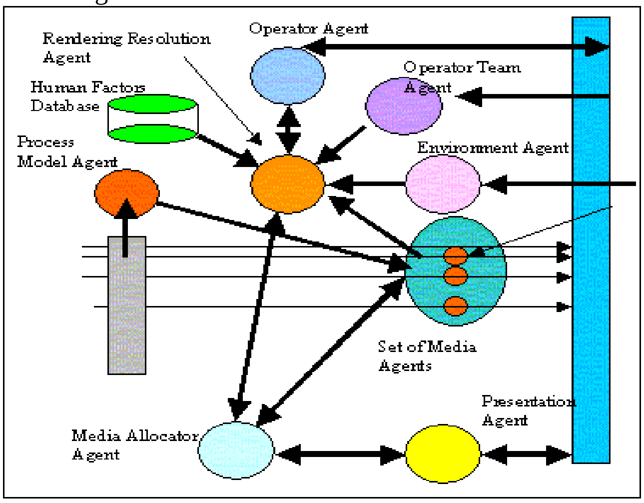

Figure 1 The AMEBICA conceptual architecture.

Each of these sources is managed by an agent, which contributes to negotiations with other agents in order to reach an agreement on the best form of adaptation.

The complete set of agents is shown in Figure 1. A *Rendering Resolution Agent*, whose job it is to query and broker the negotiations, controls the flexible mapping adaptation process. It takes into account information from an *Operator Agent* (the state of the operators), an *Environment Agent* (the state of the control room), a set of *Media Agents* (controlling presentation) *a Presentation Agent* (controlling the interface as a whole), a *Media Allocator Agent* (allocating media to information streams) and a *Process Model Agent*, which alerts the AMEBICA system to process conditions.

The *Media Allocator Agent* uses on-line reasoning to adjust the visual presentation, creating new instances of spatial layout.

9. An Example Set Of Interactions

As an example, consider the actions following a message from the *Process Model Agent* to a *Media Agent*. Let us assume that a process condition has occurred which requires the information rendered by the Media Agent to be given a much higher priority. (This example is merely one scenario and is highly simplified to illustrate a sample set of interactions.)

The Process Model detects that the condition may require system adaptation and informs the Media Agent responsible for displaying a condition of that type. The associated Media Agent then queries the AMEBICA system and requests information on an appropriate adjustment to the rendering and its parameters. To do this it informs the Rendering Resolution Agent that it has a problem, and that the problem is one of increasing priority for its object (it would probably also describe this as an alarm condition).

The Rendering Resolution Agent uses its knowledge of context to produce a list of candidate representations (from its flexible mapping list). The list is passed to the Media Allocator Agent whose job it is to select the best representation class from the list based on current interface resources usage. It does this by negotiating with the Presentation Agent, which returns information on current interface usage. This information is used to determine which of the proposed candidates is most suitable and which can be rendered at the interface.

This information is then passed to the Media Agent which implements the rendering and its parameters. If insufficient interface resources are available, the Media Allocator Agent has the power to alter the configuration of other Media Agents to allow the selected representation to be rendered. A more detailed description of the operation and interaction of the agent systems can be found in [8].

10. Spatial Adaptation Principles.

Spatial layout is unique in that it has input both from the system (*display this, highlight that*) and the operator (*move this, delete that*). The *Media Agent* picks up any operator changes to the presentation state and informs the Presentation Agent. The Presentation Agent maintains a data set containing the location and size information for each object on the screen. It also dynamically calculates weighted values for the *Available Spaces* based on the priorities of bordering windows. These weighted values are used within the system to deduce what form of spatial adaptation, if any, is necessary.

For instance, if the Windows surrounding an available space have an overall weighted value only marginally less than a requesting window, the system might decide the adaptation strategy salient for this case is to *expand* the space, by contracting the windows surrounding the space. Other options the system can take are to overlap the surrounding windows, utilise another layer, or queue the event until the necessary space becomes available. Each strategy has a difference affordance. For instance, if the requesting window is of a much higher priority than surrounding windows (therefore more important that the information is clear) the system might decide to *overlap* the surrounding windows and set the size of the window to a much greater size than would be possible by expanding the surrounding windows. . The system must also decide the appropriate size for each window. Each incoming event object has a set of constraints and data that help the system decide called *Representation Data*. For instance, every Representation Data object contains a minimum and maximum size as well as a preferred size. Thus, the system must take into account the available spaces, their location, and the preferred size of the incoming event, its priority and the weighted values of each Space. It uses these factors to determine the most effective strategy.

Satellites and Sources

As well as containing size data, each Representation Data object contain further information on the type of object to be displayed. For instance, if several windows are associated with each other, yet are called upon at different times, it is essential that the relationship between the windows be integrated within the system. For instance, if a substation equipment object is displayed and the user/system decides to open several other windows that relate to that window (for instance different measurement values for various parts of the substation), then it is preferable that the associated windows are displayed in proximity to each other. This allows the operator to easily ascertain the situation and spend less time moving windows from a standard position to a situation where they are all close together. We call the originating window (the sub station in this case) the *source* and the subsidiary windows *satellites*.

Several constraints regarding the relationship between these windows are contained within each Representation Data Object. For instance whether a window should be *near*, whether it should be *independent* (i.e. doesn't move when the source is moved) or *dependent* (*moves with source when source is moved)* as well as position information (prefer to be displayed to the left, right, above or below the source).

These factors are taken into account by the system and are used along with the size, priority, available and occupied space data to size and locate the window as the current context demands.

Parents and Children.

In Process Control type interfaces it is common to have large mimic diagrams where several objects are linked together, and yet have independent roles. For instance in a large background diagram can show how several hundred substation are connected together on a network. This diagram can be represented in several different ways, and is therefore controlled by a Media Agent that decides which representation is most appropriate at a given instance. However, the substations *within* the diagram are also controlled by Media Agents that may decide themselves to change the representation of an individual sub station (For example if a substation becomes alarmed, the system might require the substation to take on a more prominent representation). This poses a problem as we have a representation within a representation.

To deal with these we define special types of objects, *Parents* and *Children*. The parent object in this case is the large mimic diagram and the child is an individual substation. Our system allows complete flexibility in this case, as each as a child's position is fixed within the greater picture of the parent representation. However, within limits (size) it can take on whatever form of representation the system deems is most appropriate.

11. Testing of the AMEBICA Concepts

The AMEBICA Adaptability Matrix has already proved to be enormously useful in defining the adaptability conditions and responses in the two AMEBICA industrial domains. Testing of the interface has now commenced and the results will be reported elsewhere.

12. Acknowledgements

The authors are grateful to the European Commission for funding the AMEBICA project under the Basic Research initiative of ESPRIT, and wish to acknowledge the contributions made by the other partners in the project – Alcatel, Elsag, Softeco, Ecole Des Mines (Nimes), Labein and the IFE.

13. References

[1] Alty J. L. & Bergan M; "Multimedia Interfaces for Process Control: Matching Media to Tasks", *J. Control Eng Practice*, Vol. 3, No. 2, pp 241-248.

[2] André Elisabeth, Jochen Müller and Thomas Rist. WIP/PPP: "Knowledge-Based Methods for Fully Automated Multimedia Authoring. *In EUROMEDIA'96*, London, UK, pp. 95-102, 1996. Available at the URL : http://www.dfki.de/imedia/andre/

[3] André Elisabeth . WIP and PPP; "A Comparison of two Multimedia Presentation Systems in Terms of the Standard Reference Model", *In Computer Standards and Interfaces*, Volume 18, No: 6-7, pp. 555-564, 1997. Available at the URL : http://www.dfki.de/imedia/andre/

[4] Bury K.F, S.E.Davus and M.J.Darnell; "Window Management; A review of issue and some results for user testing", *Technical Report HFC-53*, IBM Human Factors Centre, San Jose, CA, June 1985

[5] Chignell M.H., P.A. Hancock; "Intelligent Interface Design", in Helander, M (ed),*Handbook of Human-Computer Interaction*, North Holland 1988

[6] Edmondson W. H. and J.F. Meech, "Putting Task Analysis into Context", SIGCHI Bulletin Vol. 26 No. 4, ACM Press ,October 1994

[7] Horvitz, E; "Principles of Mixed-Initiative User Interface", in *CHI'99 Conference Proceedings*, ACM Press 1999

[8] Khalil C.J and Alty J.L.; "The Architecture of the AMEBICA Agent Based Adaptive Process Control Interface" *Proceeding of the 18th European Annual Conference on Human Decision Making and Manual Control*, Loughborough University, UK October 25th - 27th 1999

[9] Meech J.F..; "Addressing Operator Errors in Supervisory Systems", *Proceedings of Information-Decision-Action Systems in Complex Organisations IDASCO 92*, IEE Digest 353,Oxford UK ,April 1992

[10] Opperman R.; "Adaptive User Support", Lawrence Erlbaum Associates, Hillsdale, New Jersey, UK, 1994

[11] Sullivan J.W., S.W. Tyler (eds); *Intelligent User Interfaces*, Addison-Wesley, Reading, Mass., USA, 1994

[12] Turner R. M., "Adaptive Reasoning for Real-World Problems: A Schema-Based Approach", Lawrence Erlbaum Associates, Hillsdale, New Jersey, UK, 1994

[13] Van der Veer, G.C.; "Learning, individual differences and design recommendations"; Human-Computer

Interaction. Offsetdrukkerij Haveka B.V., Alblasserdam 1990

[14] Wetzenstein-Ollenschlager E. & Wandke H; "Toward Adaptive Human-Computer Interfaces", *in Mentals Models and Human-Computer Interaction 1*, Elsevier Science Pub., North-Holland, 1990

[15] Wooldridge M. and N.R.Jennings; "Intelligent agents: Theory and Practice". *The Knowledge Engineering Review*, 10(2):115-15 1995