A Multimedia Methodology and Architecture for Computer–Aided Training and Certification

Amir Hekmatpour, Eric Millham, Larry Grant
Knowledge Based Systems Development
IBM, B21/975–1, Essex Junction, Vermont 05452
E–mail: AMIR@BTVLABVM.VNET.IBM.COM, Tel: (802)769–9627, FAX: (802)769–8444

Abstract

Existing multimedia applications are developed by media production specialists, usually delivered on read–only storage medium (CD–ROM, Videodiscs), exhibit huge initial startup cost, do not provide any intelligent control, require specialized hardware and/or software, modification and upgrades are expensive and time consuming, and are suitable for stable and often generic subject matters. This paper describes MUTANT: a multimedia shell for development of intelligent computer–aided training and certification applications. MUTANT greatly reduces the initial startup cost, allows domain experts to develop the multimedia applications without much dependency on media production specialists, delivers the applications on hard disk or removable storage medium, provides domain experts with efficient tools & techniques for upgrading applications, and provides intelligent control and management of the training and certification process. MUTANT provides the multimedia/hypermedia aspect of ExperMedia/2: an OS/2 Multimedia Expert System shell for domain experts [Hekmat93.3].

1 Introduction

This paper describes a complete and integrated methodology and architecture for development of intelligent interactive multimedia/hypermedia computer–aided training/education and qualification/certification applications. The methodology takes full advantage of modern multimedia workstations. Hypertext, hypergraphics, high resolution images, audio, motion video, and animation are technologies that in synergy are far more useful than each by itself. The modular and object–oriented architecture provides an interactive and cooperative computer–aided training and qualification development environment. The Control and Management Knowledge Bases (CMKBs) serve as an intelligent tutoring system framework.

A major departure of the approach described in this paper from previously reported computer–aided training and intelligent tutoring systems is the seamless integration of training, certification, and on–line documentation with knowledge based control and management. Existing multimedia training methodologies usually require a large initial course preparation investment and are suitable for relatively stable subject matters. Such applications usually require specialized and often expensive hardware and software. Our interactive multimedia training and certification methodology is based on standard OS/2 features and a set of novel tools and techniques and provides modular, easy to upgrade, customizable training modules without any additional dedicated hardware or software. Applications developed under this methodology and architecture reside entirely on hard disk or removable/rewritable storage medium (i.e. optical storage) and can be used in stand–alone or network configuration.

Several systems have already been developed by domain experts (i.e. process technicians, maintenance technicians, process engineers, and manufacturing engineers) using MUTANT. For example, WESDA (WESTECH wafer polisher Diagnostic Assistance)[Hekmat92], OGODA (Oxide Growth Operation Diagnostic Assistance for Thermco furnace) [Hekmat93.2], CMPKBS (Chemical–Mechanical Polish process repository), and our latest project FTDD973 [Hekmat93.1]. FTDD973 is to provide complete operator training and certification for one of our advanced semiconductor manufacturing lines (973 Fab. line).

2 Interactive Multimedia/Hypermedia Training and Education

Traditional Computer Based Training (CBT) and computer–assisted instruction (CAI) teach a subject by offering corrective feedback based on a large, pre–stored set of problems, solutions and associated remedial advice. The most noticeable problem with CBT and CAI is the amount of time it takes to produce effective courseware as all possibilities need to be anticipated in advance by the author. Some Intelligent Tutoring Systems (ITS) have addressed this problem by providing an environment where problems are generated automatically. Such "generative" systems need only be given general teaching strategies and they could produce a large number of interactions without all of them being explicitly programmed in advanced. However, most of these systems lack a clear representation of the knowledge that they are teaching. In such systems there is a mismatch between the program's internal process and those of the trainee's cognitive processes. The trainee using CBT or ITS is usually denied any chance of using his/her initiative in guiding the learning process. The CBT and CAI strategy is "learning by being told". However, research has shown that "learning by discovery" and "learning–by–doing" is in many cases a more effective strategy [Kass91] [Yazdani88]. The main goal in our training methodology has been to reduce intimidation and to put the trainee in control of learning, where it belongs. Instead of sitting in a classroom listening to an instructor, or following an operator on the manufacturing floor (who is usually fighting fires and the last thing on his/her mind is training), the trainees work individually on computers that combine text, high resolution color images, graphics, audio, animation, and motion video. This environment is characterized by student sequencing through course material in a self–paced manner, while being monitored by the system. The goals in developing the proposed training methodology can be summarized as:

– Shorten the training process.
– Improve the training quality.
– Provide consistent training across the board.
– Produce modular and portable training courseware. Since IBM is a member of the Interactive Multimedia Association (IMA), we have adapted the general principle of the IMA's practices for Multimedia Portability [IMA1] which is also known as MIL–STD–1379D Appendix D (for portable courseware).
– Effectively deliver and monitor the training objectives.
– Enable domain experts to easily develop and modify multimedia training courseware by providing them with architecture, methodology, and necessary tools and techniques.

2.1 Why Multimedia/Hypermedia–Based Training ?

The IBM Vermont plant manufactures a wide range of Integrated Circuit (IC) products (RAM, ROM, ASIC, Processors, Logic, EPROM, ...) fabricated on three full scale production lines (different technologies). Manufacturing lines are serviced by over a thousand operators, production technicians (PT), and maintenance technicians (MT). Currently, a multitude of methods are used for educating and assisting the manufacturing personnel. These methods include informal, unstructured training sessions, printed "in–house" manuals, sit–down classroom courses, and walk–thru orientations, to name a few. On the other hand, in such a manufacturing environments, the student population grows with every group of new employees (temporary or permanent). This student group is large and decentralized. They cannot all be released from the manufacturing line simultaneously to take a standard training course.

A majority of the process engineers, process technicians, and senior operators interviewed at the IBM Vermont plant, emphasized the fact that their major concern was the heterogeneous work environment they had to deal with. In addition, it was emphasized that the current training and certification methodology was inefficient and time consuming due to usual operator turn around and frequent changes in the process and tools. In most cases the operators are trained by current operators and process technicians who have their own styles and biases. In addition, most of the information required for their regular activities are textual documents (softcopy or hardcopy).

Interactive multimedia training and certification offers consistent presentation of the subject matter, on a flexible 24 hour per day seven day a week schedule. The subject matter delivered to the trainee is guaranteed to be consistent when such a methodology is used. We avoid the "Monday Morning Syndrome" when some of the subject matter presented by a line technician to new hires might be a little sketchy or missing altogether. Using such an on–line computer–based training methodology also eliminates the back–level problem. Currently, using printed documents, we're never sure if the line operator is using the most current revision or whether an operator's training had covered the latest version of process or tool upgrades. By making the information available "on–line", and integrating training, certification, and documentation, we have better control over what information is being used and what needs to be disseminated.

Most importantly, interactive multimedia training works. This technology goes beyond the point of being *user–friendly* to being *user–seductive*. People like to use this technology. This has been shown in numerous studies and we see it everyday from our contacts with existing and new users here at the IBM Vermont manufacturing plant. Thus, to use the words of Ben Franklin: "You tell me and I forget. You teach me and I remember. You involve me and I learn." We can certainly involve our trainees through the use of interactive multimedia training and certification and improve the quality and depth of the training process.

3 An Architecture For Intelligent Computer–Aided Training/Learning

The architecture proposed here provides an intelligent, interactive, and cooperative computer–aided training and qualification framework and can be classified as an Intelligent Computer–Aided Training (ICAT) or Intelligent Computer–Aided Learning (ICAL) architecture. The object–oriented modular architecture is comprised of a collection of interacting and cooperating modules arranged around the Control and Management Knowledge Bases (CMKBs), as shown in Figure 1. Training modules provide interactive multimedia training for tool, process, or business practices, as well as orientation for different areas or lines of business. Documentation modules provide on–line hypermedia documents covering all necessary information on tool, process, process parameters, logistics, and various other applications and databases which the end–user (operator, technician) comes in contact with. Certification modules provide an interactive test, evaluation, and feedback environment. CMKBs provide coordination, access control, and user management and serve as an intelligent framework for the training and certification process. CMKBs act as a liaison between the trainee and the multimedia information modules and qualification procedures. In fact, CMKBs depict a supervisor's role and provide, control, and monitor access to information and generate study plans according to the students background and progress. The user profile database includes various information on the manufacturing personnel in each area. For example, it includes employee number, name, department, tools and processes for which they are certified, date certified, level certified (*Beginner, Novice, Intermediate, Expert*), any special expertise, and authority level (system administrator, production technician, maintenance technician, operator, ...). Any module can query the user profile database directly, but the updates and modifications are managed by CMKBs or directly via GUI (Graphical User Interface) by authorized personnel (system administrator).

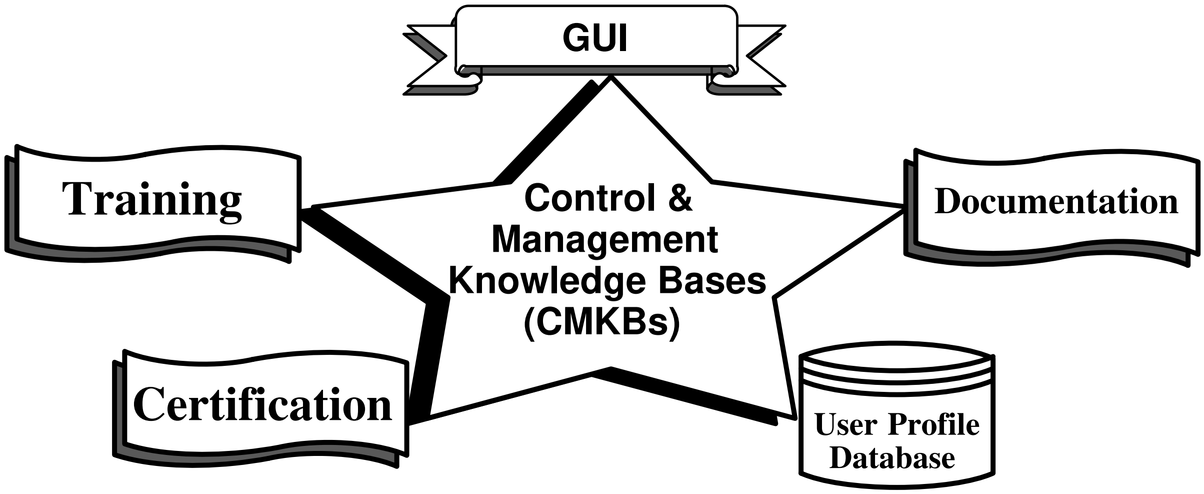

Figure 1. Intelligent Multimedia Training & Certification Architecture

3.1 Active vs. Passive Training Modules

Active training modules provide interactive step by step guidance and monitor user responses and adjust/react accordingly. These modules usually cover the more complex scenarios, where the system serves as an active participant in the process, analyzes user responses, and formulates a corresponding action based on the response and the overall goal. Active training modules could ask trainees to perform specific tasks, examine tool components, review process procedures, or make specific tool/process/product measurements before continuing to the next step. These modules are structured as implication networks or decision trees, and are implemented as Condition Action Trees (CAT) as shown in Figure 2. Active training modules are usually used for *Beginner* and *Novice* training, where the trainee needs step by step guidance, monitoring, and feedback. They are also used in dangerous situation or when dealing with hazardous material.

Passive training modules can also provide step by step instructions and procedures, but they neither enforce it, nor monitor the progress. Such modules are suitable for *Intermediate* and *Expert* users who may not require step by step control and guidance and do not have to be monitored as closely as *Beginner* and *Novice* users. Whenever multimedia and hypermedia modules are accessed directly, they provide passive training, whereas if these are presented via diagnostic knowledge bases or the CMKB, then active training is performed.

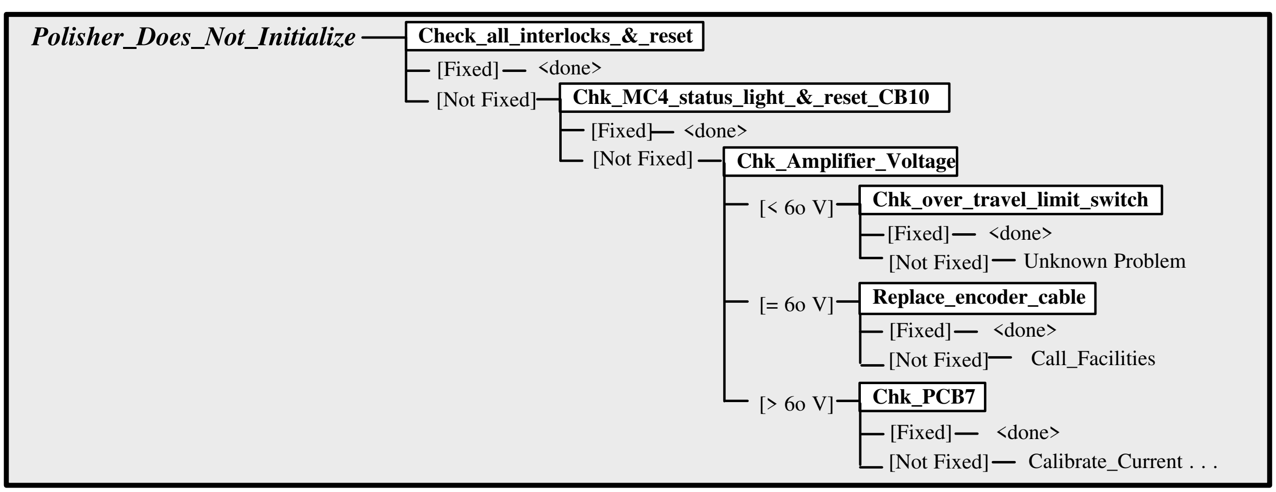

Figure 2. Condition Action Tree (CAT) for Active Training

4 Multilevel Student Model

Computer–based training and intelligent tutoring systems are usually based on a single and simple student model. Regardless of the familiarity or lack of familiarity of the trainees with the topic, all go through the same training process. A new hire who has never worked in a similar environment is given the same training material as the one who has many years of related experience and might have been transferred to this new assignment from a similar area. Certification and qualification procedures are usually conducted orally in an ad–hoc fashion and are subject to the trainer's judgment and biases. In addition, there is no formal methodology for increasing the responsibility of a trainee as the training proceeds. In order for a computer–aided training system to acquire the necessary flexibility, it should distinguish between several types of students and structure the training material according to the students needs and background.

To address this problem and to provide a methodology for presenting and managing the training material to students of varying background and expertise, we have developed a new multilevel student model. This new model advances the trainee's rank as the training progresses and as the trainee improves his/her certification test score.

Expert PTs, MTs and operators were interviewed to identify efficient methodologies for training the manufacturing personnel in their area. In other words, how should we present and manage the training material to students of varying background and expertise? Four levels were identified; *Beginner, Novice, Intermediate*, and *Expert* (Figure 3). A trainee rated as a *Beginner*, would require the most assistance and step by step guidance, Whereas an *Expert* would only need to be able to locate the information rapidly and efficiently as the need arises. A student/trainee is assigned an initial level by the area PT. The student advances to the next level (increasing responsibility and access to information) as he/she passes the certification procedure for that level. As the user level is advanced, the detailed procedural information is reduced, but the conceptual and deep knowledge (declarative and procedural knowledge) of the process and the tool is increased. A *Novice* student knows the steps involved in the operation, an *Intermediate* student should know the cardinality and temporal relationship and importance of the steps. An *Expert* user would also understand the taxonomy of the tasks and procedures.

Such a multilevel student model reduces the danger of trainees getting involved in potentially dangerous or destructive tasks which they may not have had adequate training and preparation for. In the existing training methodologies, once a trainee is certified, he/she could be assigned to process products although he/she may not have had any training in process or tool troubleshooting. In most cases, ignoring initial signs of problems or not being able to respond to problems quickly, could be very costly and result in tool or process shut down. On the other hand, in some situations, initiatives by operators who have not been adequately trained and informed of their responsibilities could result in scraps, reworks, damaged tools and products, or an out of spec process. In our multilevel methodology, CMKB modules keep track of the user level and last certification date, and schedule training sessions and topics. In addition, these modules recommend tasks suitable for each operator and decide whether the operator should have access to process and/or tool diagnostics.

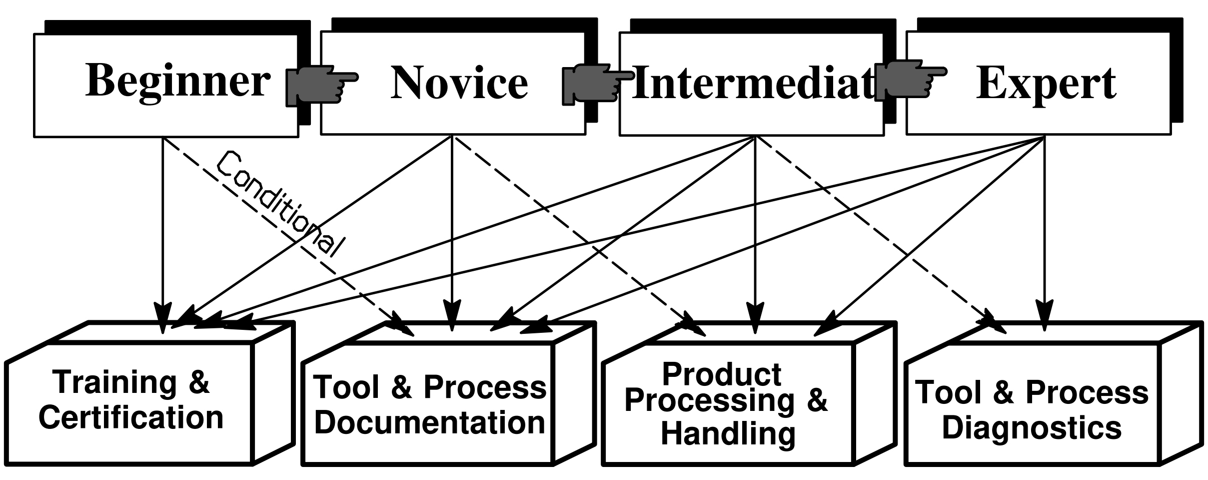

Figure 3. Multilevel Training and Certification Model

4.1 User Level Management and Control

User name, department, passwords, user level, process and tools certified for, certification dates, authorities, and specialized expertise are maintained in the user profile database for each user. User level is identified and assigned to the global environment variable USER_LEVEL at logon time, but may be changed at anytime during the session (e.g. after the operator takes a certification test). Global access to training, documentation, logistics, and diagnostics is controlled by CMKBs, whereas local access to specific sections of training modules or on–line documents are controlled via local environment variables which are maintained by the corresponding module. The value of these local environment variables determines which sections of a module can be accessed by a user and which sections should be hidden. In the example shown below, the first section will be available to *Expert* users and hidden from all others, section 2 will be shown to both *Beginner* and *Novice* users, and section 3 is shown to all but *Beginner* users.

:hide user_level='Expert'
:h1.Section 1
:p. procedures for experts
:ehide.
:hide user_level='Beginner + Novice'
:h1.Section 2
:p. Information suitable for beginner and novice operators
:ehide.
:hide user_level=' Novice + Intermediate + Expert'
:h1.Section 3
:p. This information is hidden from Beginner users.
:ehide.

5 Multimedia/Hypermedia Modules

Multimedia documents and training modules are designed and developed using C, Presentation Manager (PM), Dialogue Manager (DM), OS/2 Information Presentation Facility (IPF), and IPF compiler (IPFC). Each section in the document is assigned a unique id (eg. *:h1 res=100.* Polish Arm). Then any word, sentence, section, image, hot–spot in an image can be linked to this section using a link tag (*:link reftype=hd res=100.*click here to review Polish Arm section*:elink.*). Once the document is compiled, the text "Click here to review Polish Arm section" is uniquely highlighted so the user knows it is selectable. Pictures can also be embedded into the document as artwork (*:artwork name='arm.bmp' linkfile='arm.hgl'.*). The linkfile contains the description and coordinates of hot–spots defined for the image. Each hot–spot can have up to 4 functions associated with it, which can be selected from the hot–spot functions pop–up menu, as shown in Figure 4. The hot–spot functions menu is selected by clicking on the object of interest. The hot–spot functions are in fact secondary hot–spots within the primary hot–spot window. These functions are links to other parts of the document (show text, display close–up picture) or to external applications to show an animation, display a video clip, or play audio (e.g. *:link reftype=launch object='audio.exe' data='calibrt.aud'.*Click here to listen to calibration procedures. *:elink.*).

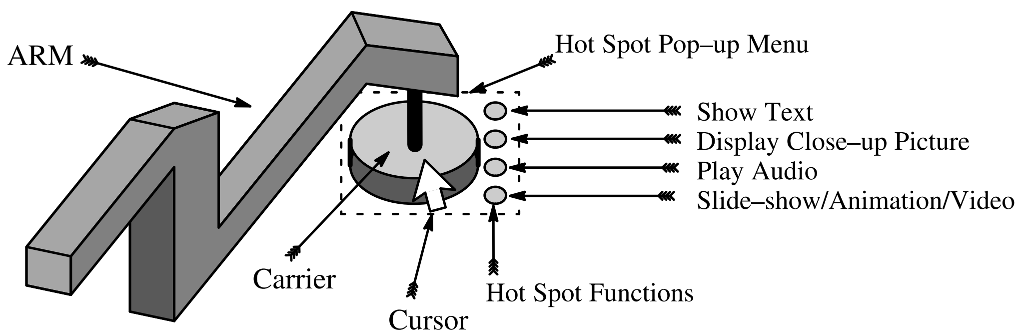

Figure 4. Hypergraphic Hot–Spot and Hot–Spot Functions

5.2 Partitioning of Multimedia Documents into Logical Pages

Coordinating text, graphics, and images in a multimedia environment is very important in effective presentation and management of the subject matter [Feiner90,Hekmat92,Hekmat93.1]. Some researchers have investigated the automatic generation of coordinated multimedia [Feiner91]. Our methodology is based on a pre–defined presentation scenario and is implemented in one of the pre–defined templates. Multimedia documents are partitioned into "Logical Pages". A logical page (Figure 5) consists of a set of images and all their associated textual description. In other words, a logical page is the predefined collection of all related information (text, graphics, images, audio, animation, video clips, ...) which the trainee should/could review when studying the subject matter covered in that page. The appearance, format, and access to logical pages are fixed and consistent throughout the system. Logical pages are related to each other via hypertext and hypergraphic links. To accommodate all possible situations and to maintain consistency among various multimedia modules developed in an organization, a set of "Logical Page Templates" are defined (Partial list shown in Figure 6).

5.2.1 Logical Page Templates

In order to provide consistency among various multimedia courseware in an organization and to provide a library of templates for application developers, a set of logical page templates have been defined. These provide consistent facilities for embedding various multimedia and hypermedia capabilities in documents. Figure 6 shows a partial list of suggested page templates. These templates facilitate sharing of source code between various modules within a project. In addition, multimedia documents can be shared among projects, maintaining a consistent user interaction and presentation.

Each logical page has a primary window which is considered parent and owner of all other windows. All windows in a logical page have a unique group number(group=1 – 64000). Only one window of each group can be opened at any one time. IPF searches among the open windows to find one with a number matching the one specified with group=. If no match is found, IPF opens a new window. If a match is found, the information associated with the group number is swapped with the information in the matched window. In addition, each window has a unique window identifier (res=1 – 64000) which is used for hypertext and hypergraphic linking to that window, and a alphanumeric identifier (id=alphanumeric). The primary window identifier is considered the logical page number (suggested primary window identifiers, res= 0, 10, 20, 30, 40, ...) and each logical page can have up to nine secondary dependent windows (0–> 1,2,3, ..., 9 and 10–> 11, 12, 13,...,19). Although, there is no linking to secondary windows in our logical page architecture, we still assign res= and id= to each window for identification and organization purposes.

```
:h1 res=0 id=primary1 group=1.First Primary Window
:p.Some text
:h2 res=1 id=secondary1 group=2.First Secondary Window
:artwork name='image.bmp' linkfile='image.hgl.
:h2 res=2 id=Secondary2 group=3.Second Secondary Window
:artwork name='chart.bmp fit.
```

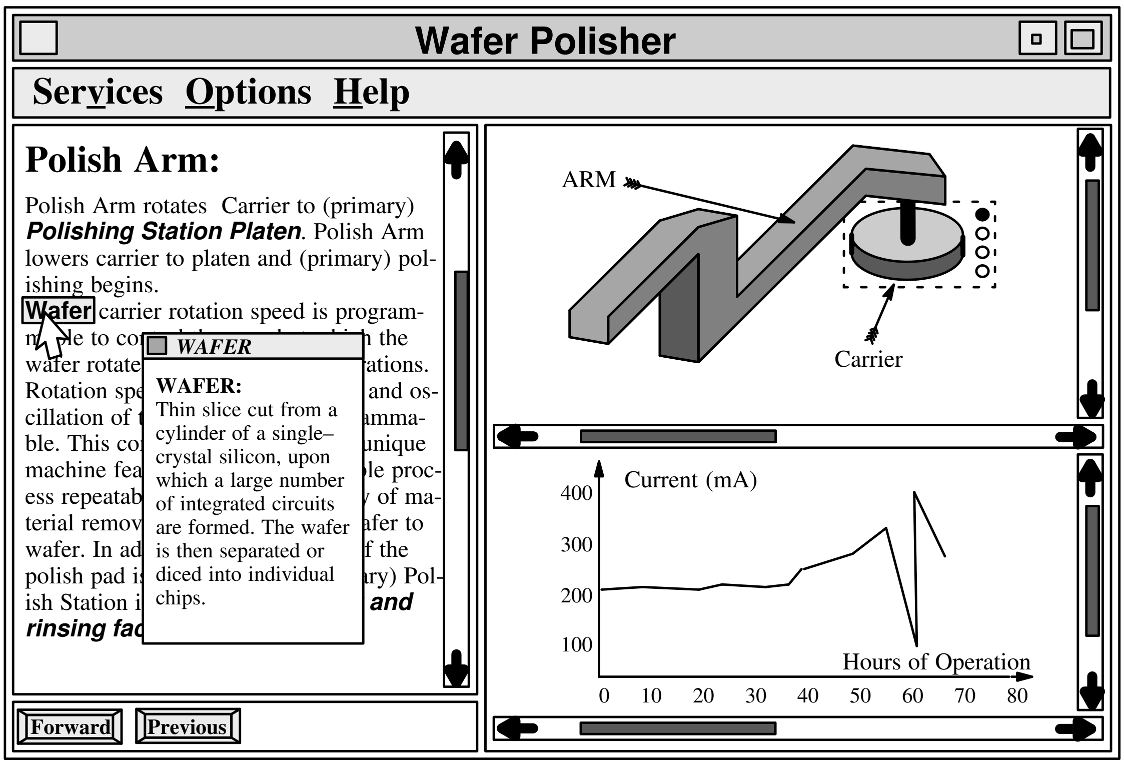

Figure 5. An Example of Multimedia Document Logical Page

To assure a complete and accurate display of a logical page (regardless of how it is addressed), it can only be linked to via its primary window (group=1 in Figure 6). A logical page, similar to a page of a book, may contain many different objects of interest, but can only be opened by a call to its primary window. All other windows in a page are either dependent and automatically linked to the primary window (group=2,3,4) or are pop–up windows which are activated via hypertext or hypergraphic links (group=5,6). When a primary window is opened (via hypertext link, hypergraphic link, selection from table of contents, external call, or as a result of a search) all its dependent secondary windows are opened automatically. The following is the source code for the logical page shown in figure 5.

```
:userproc.
:docprof toc=1 dll='page_atr' objectname='page_attribute_main' objectid='file.opt'.
:title. Wafer Polisher
:h1 res=10 x=left y=bottom width=40% height=100% group=1 titlebar=none scroll=vertical rules=none. Polish Arm
:link reftype=hd res=11 group=2 vpx=right vpy=top vpcx=60% vpcy=50% auto dependent ctrlrefid=bottom1.
:link reftype=hd res=12 group=3 vpx=right vpy=bottom vpcx=60% vpcy=50% auto dependent.
:p.:hp2Polish Arm:ehp2.
:p.Polish Arm rotates Carrier to (primary) :link. reftype=hd res=20.Polishing Station Platen:elink.. Polish Arm lowers carrier to platen and (primary) polishing begins.
:p.:link refid=wafer reftype=fn.Wafer:elink. carrier rotation speed is programmable ...
...
:h2 res=11 ctrlarea=page hide nosearch noprint titlebar=none rules=none.
:artwork name='arm.bmp' linkfile='arm.hgl'.
:h2 res=12 ctrlarea=none hide nosearch noprint titlebar=none rules=none.
:artwork name='chart.bmp'.
```

:fn id=wafer.
Thin slice cut from a cylinder of a single–crystal silicon, upon which ...
:efn.
...
:euserdoc.

A logical page always has a text window which serves as its primary window, except full screen non–textual pages (image, graphics, audio, motion video, animation, external applications) in which case the page title is considered the textual window. A window can be assigned vertical and horizontal scrolls, as well as title, background and foreground colors, font style and size. It is recommended that text windows be assigned only vertical scrolls and allow IPF compiler to fit and wrap the text in the window.

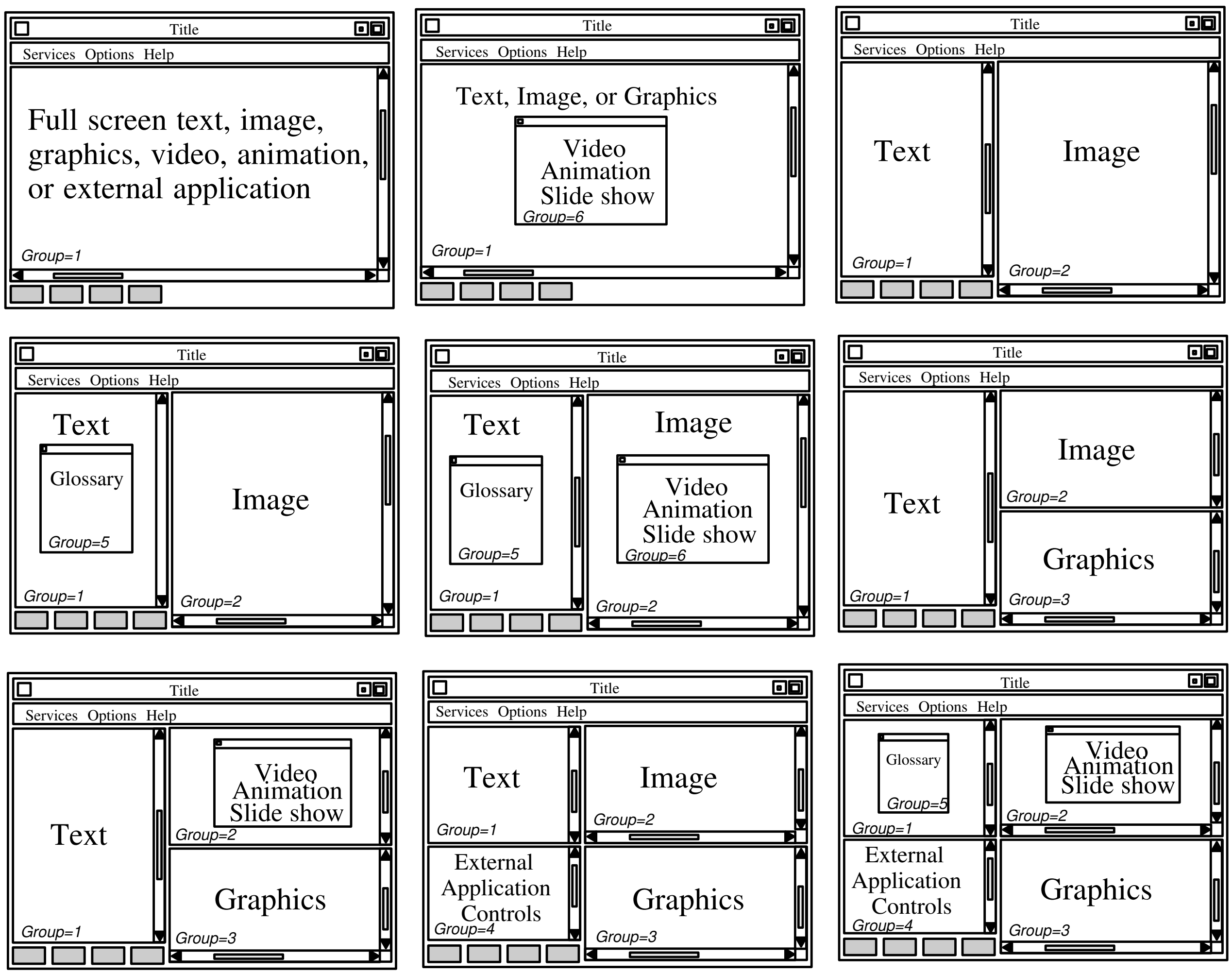

Figure 6. A Partial List of Multimedia Logical Page Styles

5.2.2) Customizing the Logical Page

Very early in our field testing of some of the multimedia applications developed in MUTANT, we realized that many of the trainees are not sophisticated computer users and too many features (options, menus, controls) may be confusing to them and in some cases restrict the amount of control and access that CMKB modules need to preserve. For example, it was noticed that if the users are allowed to minimize training applications, the system would not be able to control the number of copies opened and would not know whether the task was completed or not. In some cases users could not locate a document after it had been minimized. In addition, to be consistent with our multilevel student model, it was felt that not all users should have access to all action bar menu options (i.e. copy, print, bookmark, ...). Therefore, we developed a utility which allows the developer to customize logical page attributes.

In addition to the logical page templates described in previous section, most of the major attributes of a logical page can be customized. For example, page background color, highlight color, logical page size and position, text style and font size, pull down menus and menu items, system menu items, push buttons, vertical and horizontal scrolls, min/max icons, window titles, etc. Some of these are controlled via IPF tagging language (e.g. *:font facename=Courier size=13x8.* or *:color fc=yellow bc=black.*) and some are controlled by utilities provided by MUTANT. For example, Figure 7 shows the front–end of the

Page_atr utility which allows the user to customize the logical page attributes.

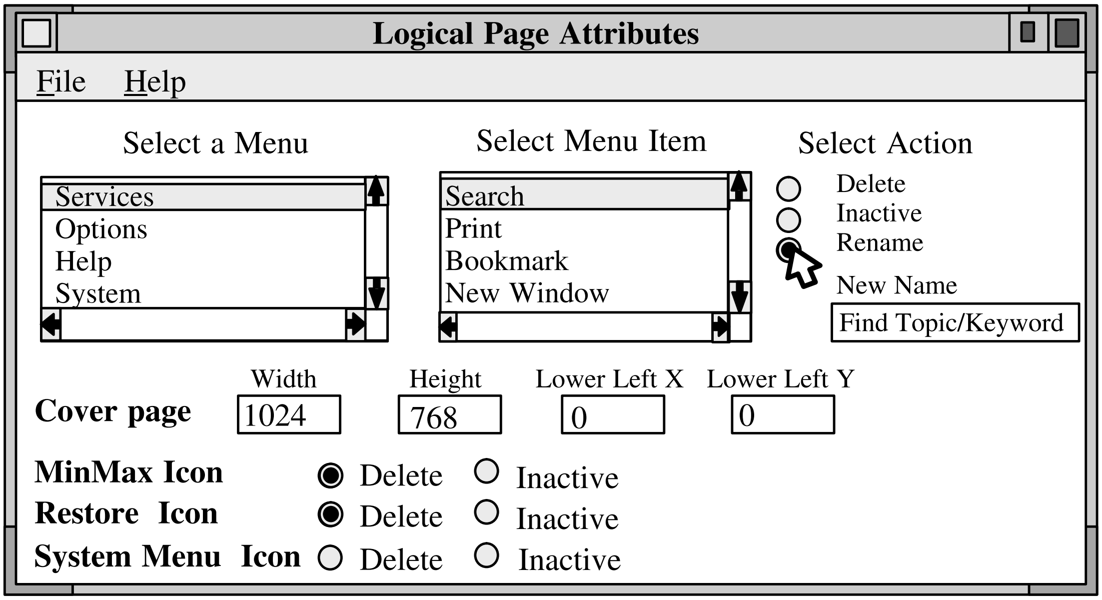

Figure 7. Logical Page Attribute Customization Front–End

6 Motion Video and Animation

Our studies have shown that multimedia in general, motion video, animation, and audio in particular are very appealing to manufacturing personnel involved with maintenance, diagnostics, and training. However, in many cases a major concern was the cost and logistics of the required hardware and software. We have to remember that most manufacturing functions include tens and even hundreds of workstations. If an application requires special hardware or software license, the cost would become a major roadblock. Although cost was the major concern, another problem was the lack of any available expansion slot on some workstations in the area.

6.1 Poor Man's Video: A programmable video generation software shell

Each manufacturing line consists of hundreds of programmable workstations. Training, certification, and on–line documents have to be installed on a large number of these workstations. We realized that it is not practical to expect or justify upgrading all of these workstations with video card (e.g. IBM's M–motion adapter card for motion video), Video capture adapter (for digitizing and storing of compressed video images), and the Audio Capture and Playback Adapter (ACPA). It was also felt that it is essential for all operators and technicians to have access to the training, certification, and documentation applications on the workstations dedicated to their own workplace, rather than on some workstation located away from their workplace. Therefore we started looking into alternative methods for generating and playing motion video, animation, and audio.

We had already developed an OS/2 PM program for displaying bitmap slide shows. It basically loaded and displayed OS/2 bitmap images according to a predefined order and delay. We decided to take this concept one step further, add frame delay control capability, improve the performance and integrate it with IPF's communication object concept. The resulting package which we have nick named "Poor Man's video (PMV)" is written in C and PM and can be called from IPF documents as an application–controlled window, as shown below;

:h1 res=999 id=video1 x=center y=center width=400x height=300x scroll=none titlebar=sysmenu rules=none hide group=6 Rapitran Loading/Unloading Procedure
:acviewport dll='pmv' objectname='pmv_main' objectid='c:\rapitran\bitmaps'.

PMV allows the user to specify a set of OS/2 bitmaps as video frames and specify a global frame delay (1–99999 ms) or specific frame delays for each frame or a group of frames (Figure 8). These bitmaps can be either images captured with a still or video camera and converted to OS/2 BMPs, scanned images, screens captured by PMCAMERA, or graphics generated with graphic programs such as IBM's CUAPAINT or PMDRAW. PMV generates a DLL which is loaded to an IPF communication object window, serving as the video frame. We have been able to achieve about 45 frames per second (256 color 300x400 pixels frame) on a model 95 PS/2 with 32 Meg. RAM, with an initial DLL load time of 7 seconds. The PMV's graphical front–end (Figure 8) allows application developers to create video and animation clips quickly and efficiently.

6.2 Run time vs. Compile time DLLs

PMV can generate either a "run time" DLL or a "compile time" DLL. Compile time DLLs are self contained and can be exported to other workstations without any need for copying the images or having access to them. Run time DLLs, on the other hand, contain pointers to the images used in the video clip or animation and therefore require access to these. In some cases images are used in more than one animation or video clip. Run time DLLs are more efficient in such cases since these are generated and loaded at the run time, as such, a single copy of the frames to be used in animation or video clips need to be stored. When an animation or a video clip is launched, the run time DLL loads the images on the fly, which could be time consuming for long video clips (about 15 seconds for a 60 frame clip on a model 95 PS/2). In addition, the launch would fail for run time DLLs if the images can not be located (moved or renamed). On the other hand, there is the storage overhead associated with compile time DLLs, since these have to be replicated and stored locally (accessing large DLLs over the network is slow and unpredictable). We are currently looking into compression algorithms which might be applicable to our PMV DLLs.

6.3 Animation

PMV can be used for pseudo animation using graphic bitmaps or still images with 500 ms to 1 second of frame delay time. In addition, we can also use a true animation tool called AnDes which allows the user to animate models created using Excelerator. Models of mechanical, electrical, or electro–mechanical components and tools can be created following Ward & Mellor methodology, which is based on transformation graphs and state transition diagrams. Excelerator models are then converted to AnDes format, before they can be animated. Although AnDes offers a true animation capability, in most cases our developers prefer to use PMV and avoid the overhead associated with installing Excelerator, AnDes, and Prolog for creating and performing animation.

7 Interactive Certification and Qualification

The certification process is interactive and cooperative and is geared towards process and tool operator certification in general, and semiconductor manufacturing operator certification in particular. The certification procedures test the trainee's progress, track his/her performance and help the trainee to concentrate on the problem area. The feedback and tracking was developed in response to the concerns with the existing certification and training procedures which lacked a meaningful and unintimidating feedback mechanism. In addition, to reduce intimidation, enhance learning and increase the trainee involvement, we expanded the certification beyond a simple pass or fail. Certification modules can be considered an extension of the training by providing meaningful feedback, suggesting topics to be reviewed, and providing general comments as well constructive criticism.

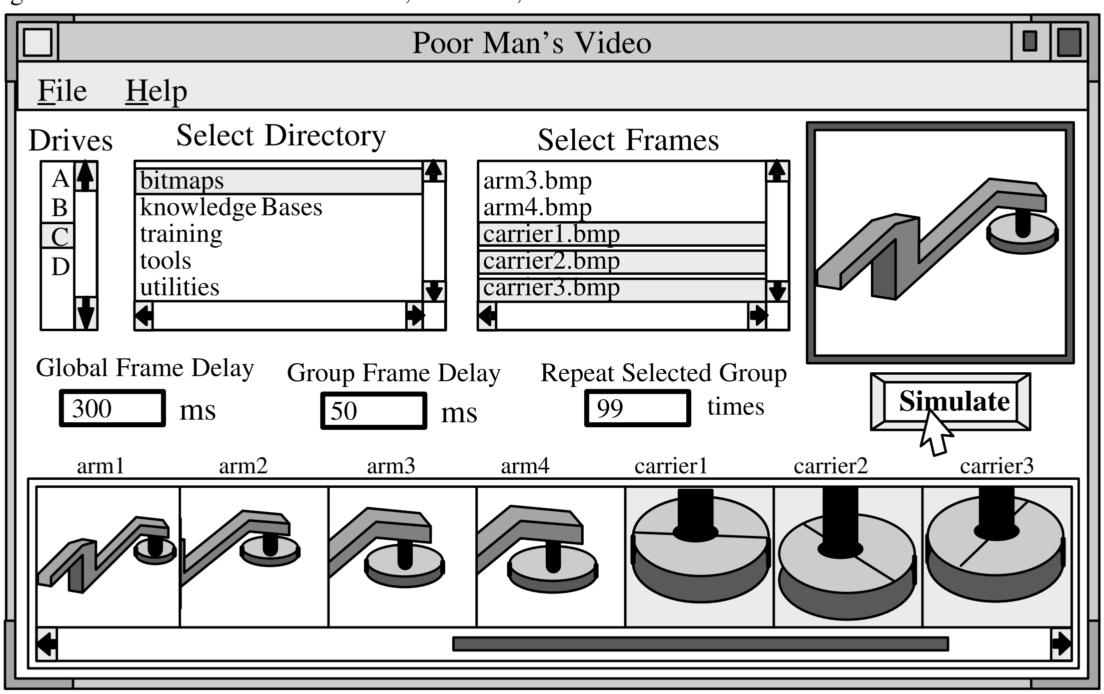

Figure 8. Poor Man's Video Front–End

A new user needs a password and a valid IBM employee serial number to take a certification test. New users are assigned USER_LEVEL="Beginner", unless a different level is assigned to them by the PT or MT responsible for the area. Once a trainee has reviewed all the required materials and procedures for his/her area, tool group, and level, he/she will be given an on–line certification test. The test is usually comprised of about 20–50 questions (multiple choice and True/False) which may be answered in any order. The trainee can review an up–to–the –point status summary which shows the user responses to questions and whether a question has not been visited yet (Not Answered) or no answer has been selected for it yet (skip). Once the test is turned in, the CMKB evaluates the responses and generates a report. For each question answered incorrectly, the system lists topics, documents, or training modules to be reviewed.

8 Hypermedia On–line Documentation

Tool operating and maintenance manuals, Process documents, computer system user guides, safety and emergency procedures, and hazardous material handling and disposal procedures are converted to hypertext documents and made available as part of the on–line documentation. These documents are accessible from training and certification modules, as well as directly from the system front–end. External applications (e.g. diagnostic expert systems) can also utilize these documents. A document can be opened up to a specific topic or section. All documents may also be searched for a keyword or topic prior to opening. In case of multiple matches, it opens up to the first match, with the section title of all matched cases displayed in a selectable list box. A hypertext maintenance log can also be set up, so all activities related to the tool, process, or business practice are logged in for shift–to–shift communication and long term documentation. In addition, periodic and long term maintenance activities, tool shut–downs, as well as technician/operator vacation plans can be scheduled and posted on this log.

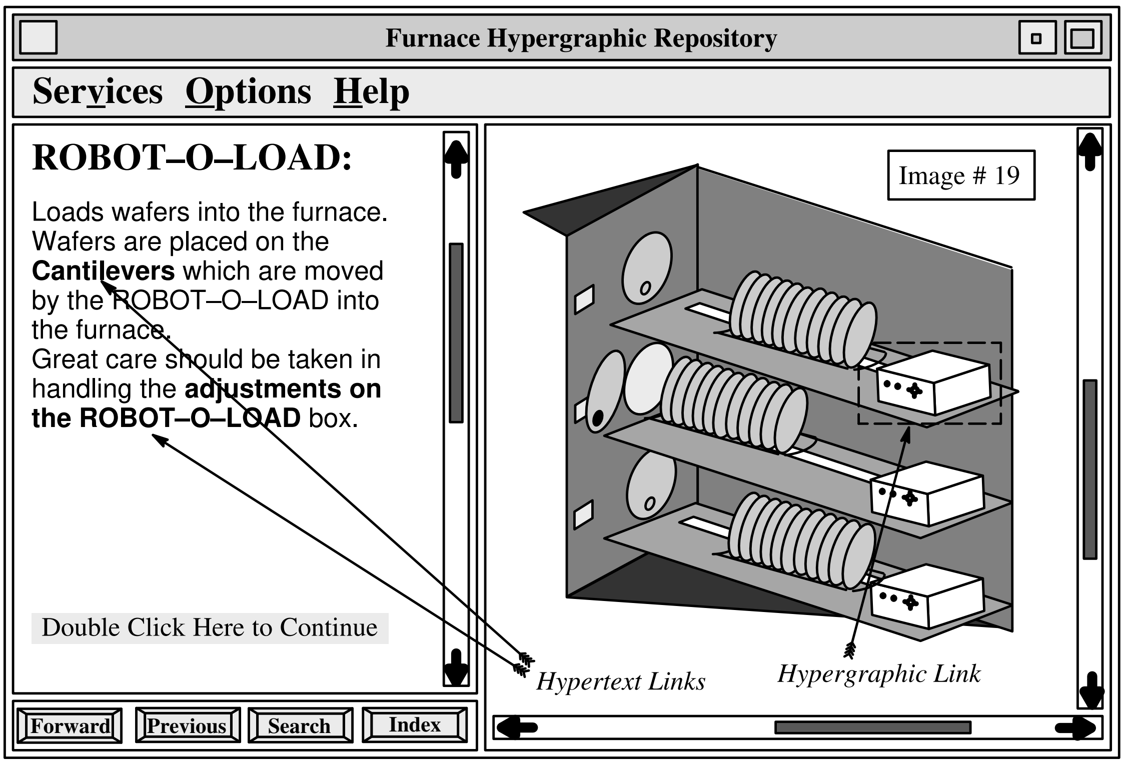

Figure 9. An Example of Hypergraphic Repository Logical Page

8.1 Hypergraphic Repository

As the development of multimedia applications proceed, soon the developer is faced with storing, organizing, and managing hundreds of still images. The multimedia documents, slide shows, animations, and video clips utilize these images. In addition, most of these images contain objects (components, tools, instruments, ...) which might be of interest during development as well as during training.

To address these requirements MUTANT provides utilities to develop hypergraphic repositories. The repository is more than a database of still images and animation and video clips. It combines various types of information about the images, contents of images (objects of interests in individual images), their cardinality, and temporal relationships in one place. The repository serves as one of the on–line documents and is accessible from training, certification, other on–line documents, as well as directly from front–end. The repository is comprised of logical pages, where each page contains an image window (high resolution picture, video clip, animation graphic) and a textual description of the image, its contents, or topics and procedures conveyed by the image, as shown in Figure 9. The image window is a dependent child of the text window (main window for the logical page) and is auto linked to the main window. The interesting objects in an image (e.g. important components of a tool) have simple hot spots assigned to them. These hot spots link the object to other logical pages where additional and specific information is provided. The repository also provides pull–down menus for topic, keyword, and object query and selection.

A database containing description and whereabouts of the images is also associated with the repository. This database is created and maintained in the OS/2 database manager and is available to application developers as well as the end–users. The database maintains information on contents of the image, where it is physically stored, and how to down load and/or display it. Each image in the repository is assigned an id (i.e. image19, displayed in the upper right corner of image window, Figure 9) which is the index under which the image information is stored in the database.

9 CMKBs & Certification User Interface

Graphical user interfaces (GUI), multiple screen displays, and windowing capabilities have provided powerful possibilities for user interface design in recent years. Many researchers in the field of human–computer interaction and cognitive science have extensively studied the effects and requirements of such an interface environment on users as well as system design [Card84, Bury85, Gaylin86]. With the advent of windowing and multiple screen displaying capability, additional issues and concerns have to be taken into consideration in design of computer user interfaces. Our earlier work on the design of user interfaces for expert systems, investigated the requirements for management and effective presentation of technical information in window–based graphical user interfaces [Hekmat90]. The user interface architecture described in this section is adopted from this work.

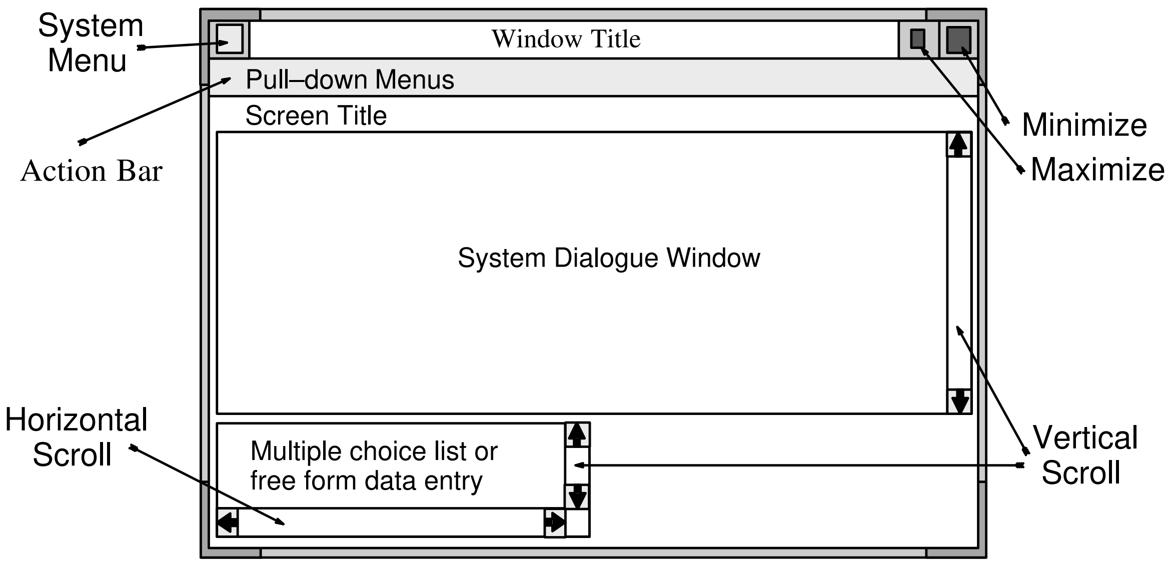

Figure 10. CMKBs and Certification User Interface Template

The user interaction with CMKBs and the certification process is through a graphical, window–based interface consisting of visual representations and interaction facilities (Figure 10). Interaction with the system is through a menu–driven, point and click interface. Manual data entry and typing are kept to a minimum. The user interface methodology is based on IBM's Common User Access (CUA) architecture [IBM1]. CUA is a set of guidelines to be used to design user interfaces that are easy to learn and use. The CUA architecture defines the workplace environment for programmable workstations. These guidelines are based on user–interface and object–oriented relationships and have been supported by human factors research and usability testing. They specify common user–interface components and techniques, as well as guidelines for applying them [IBM2]. The pull–down menus and system menu are consistent with OS/2 Presentation Manager standards and the IBM CUA guidelines, as shown in Figure 10.

9.1 CMKBs & Certification Window Structure

In addition to the logical page styles described for multimedia documents, a consistent windowing structure is adapted for interaction with and presentation to the end–user during login, session management, certification, and other potential applications (i.e. diagnostics, process control, tool setup, recipe compile, etc.). The computer monitor screen is divided into four sections (Figure 11). The "system window" (upper left) contains the system dialog, action bar menus, options, textual information to be displayed to user, and the standard PM service icons (Minimize, Maximize, Scroll up, Scroll down, and system menu). The "summary window" (lower left) appears at the end of a certification session or when a conclusion is reached. Warnings, notices, and important messages are displayed in the warning window (upper right) in a bright color to get attention. All graphics, digitized images, pictures, diagrams, slide shows, animations, and motion video clips appear in the "graphics window" (lower right). If there are more than one picture to be displayed, multiple "graphics windows" are opened on top of each other with a 20 pixel off–set towards the center of the screen. Any "graphics window" can be brought up to the front by just clicking on any exposed part of it. All windows are opened and closed by CMKBs during the session, but the user can also move/size/minimize/maximize or closed any of the windows (except the system window) at any time.

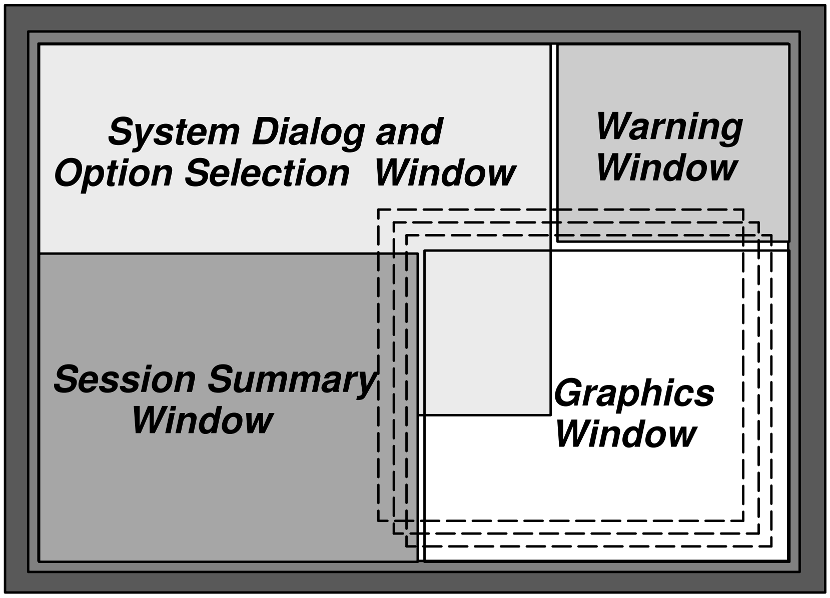

Figure 11. Diagnostics and Certification User Interface Window Structure Template

10 References

[Bury85] Bury K. F., Davie S. E., and Darnell M. J., “Window Management: a review of issues and some results from user testing, Technical Report HFC–53, Human Factors Center, IBM San Jose, California, June 1985.

[Card84] Card S. K., Pavel M., and Farrell J., “Window–based computer dialogues,” Proc. the INTERACT84, First IFIP Conference on Human–Computer Interaction, London, Elsevier, Amsterdam, The Netherlands, pp. 355–359, 1984.

[Feiner90] Feiner S. and McKeown K., ”Coordinating Text and Graphics in Explanation Generation,” Proc. AAAI–90, pp. 442–449, Boston, MA, July 29–August 3, 1990.

[Feiner91] Feiner S. and McKeown K., ”Automating the Generation of Coordinated Multimedia Explanation,” IEEE Computer 24(10), pp. 33–41, October 91.

[Gaylin86] Gaylin K. B., “How are windows used? Some notes on creating an empirically–based windowing benchmark task,” Proc. Computer–Human Interaction (CHI86) Human Factors in Computing Systems, pp. 96–100, New York, 1986.

[Hekmat90] Hekmatpour A, Vassigh A., and Norman D. A. “Conceptual Modeling for Expert System User Interface Development,” Proc. 7th Int. Association of Science and Technology for Development (IASTED) Int. Symposium on Expert Systems Theory & Applications, Long Beach, California, December 1990.

[Hekmat92]A. Hekmatpour, Charles Elkan “A Multimedia Expert System for Wafer Polisher Maintenance,” Technical Report #CS92–257, Dept. of Computer Science and Engineering, University of California, San diego, August 1992.

[Hekmat93.1]]A. Hekmatpour, G. Brown, R. Brault, G. Bowen, L. Grant, E. Millham, “FTDD973: A Multimedia Knowledge Based System and Methodology for Operator Training and Diagnostics,” Proc. of Conference on Intelligent Computer–Aided Training and Virtual Environment Technology (ICAT/VET–93), NASA/Johnson Space Center, Houston, Texas, May 1993.

[Hekmat93.2]A. Hekmatpour, G. Brown, R. Magnuson, “OGODA: A Multimedia Knowledge Based System for Thermco Furnace Diagnostics & Operator Training,” Technical Report # 1993.47, IBM Vermont, June 1993.

[Hekmat93.3] A. Hekmatpour, E. Millham, L. Grant “ExperMedia/2: An OS/2 Multimedia Expert System Shell for Domain Experts,” Proc. World Conference on Educational Multimedia and Hypermedia Conference (ED–MEDIA 93), Orlando, Florida, June 1993.

[IBM1] IBM Document Number SC34–4289, “Systems Application Architecture Common User Access Guide to User Interface Design (CUA Guide),” 1989; Available through IBM branch offices.

[IBM2] IBM Document Number SC26–4582, “Common User Access Advanced Interface Design,” 1989; Available through IBM branch offices.

[IMA1] Interactive Multimedia Association, ”Recommended Practices for Multimedia Portability,” Version 1.1, 800 K Street, N.W., Suite 240, Washington, D.C., 20001..

[Kass91] Kass A. and Blevis E., ”Learning Through Experience: An Intelligent Learning–by–Doing Environment for Business Consultants,” Proc. of Conf. on Intelligent Computer–Aided Training (ICAT91), NASA Publication 10100, pp.289–302, 1991.

[Maybury91] Maybury M. T., ”Planning Multimedia Explanation Using Communicative Acts,” Proc. of AAAI–91, pp.61–66.

[McKeown92] McKeown K., Feiner S., Robin J., Seligmann D. D. and Tanenblatt M., ”Generating Cross–References for Multimedia Explanation,” Proc. AAAI–92, pp. 9–16, San Jose, CA, July 12–16, 1992.

[Yazdani88] Yazdani Masoud, ”Expert Tutoring Systems,” Expert Systems, Vol. 5, No. 4, pp. 270–271, November 1988.