A Multimedia Information System For IC Failure Analysis

S. C. Hui[1], A. Goh[1] and L. H. Lau[2]

[1] *Division of Computer Technology, School of Applied Science, Nanyang Technological University, Singapore 2263*

[2] *National Semiconductor, 11 Lorong 3, Toa Payoh, Singapore 1231*

An object-oriented approach for constructing an IC failure analysis system environment named MultiICFA is described. The MultiICFA enables the failure analyst to manage IC analysis information, which consists of video, graphical plot, film, image, and text description, through a user interface. In this paper, the characteristics of failure analysis in an IC device are described and a process of failure analysis, which is currently used by an IC manufacturer, is given. The architecture of the MultiICFA is outlined, which focuses on modelling and management of data. An object-oriented approach to design is adopted in the implementation of the system. The goal of the system is to enable the failure analyst to formulate event-driven queries through the user interface in order to display the test results of faulty IC devices.

Received October 1992

1. INTRODUCTION

Failure analysis of an IC is often performed by a manufacturer under two circumstances: when a newly manufactured IC device fails to work, and when an IC device fails at the customer site. Failure analysis is important not only for determining the reasons why an IC device fails, but also for understanding how an IC device fails. It enables the IC manufacturer to find out whether it is a design fault or a manufacturing fault. The IC manufacturer will gain a competitive edge if IC device faults can be diagnosed as early as possible. In current IC manufacturing environments, the need to have a fast response time in trouble shooting and problem solving has resulted in failure analysis playing an increasingly significant role. A team of failure analysts are thus employed to perform failure analysis on particular IC devices through a series of nondestructive and/or destructive tests. Highly specialised equipment is used to inspect different parts and characteristics of the faulty IC device. An X-ray is used to check the integrity of bond wires and Scanning Acoustic Microscopy is used to check any package cracks, delaminations and voids in the device, from which photographs and video signals can be taken. The failure analyst has to review all these test outputs which are presented in different forms and formats such as video, photographs, graphical plots and text descriptions. After inspecting the test output, the analyst will comment on the observed anomalies and mark them on the corresponding output media. Conclusions are drawn based on the observed results. Finally, a report is written for the tested IC device for management review, submission to the requester and filing.

Although most of the test output can be electronically-processed and stored, the IC manufacturer has adopted a paper filing system to store all the results and conclusions of the tested IC devices. Different forms of output and data for each tested IC device are collected and filed. Only the summary is maintained in a relational database system which is running on a personal computer.

In addition to taking up storage space, the process of using a manual filing system is time consuming. It is also cumbersome as users have to search through the stacks of files to locate the desired detailed information of individually tested IC devices for reference. As time goes by, more IC device test records are created and the size of the filing system grows.

In order to meet the challenge of today's IC manufacturing environment, it is necessary for some form of automation in the analysis procedure and management of data to be adopted. The work described in this paper concentrates on the data management aspect. Failure analysts can make a better diagnosis if the complete set of results from a tested IC device is available for immediate access under an analysis environment. This will help the failure analyst to work more efficiently and at the same time, eliminate mistakes committed due to a lack of information. In addition, a database of tested IC devices can also be made available. This enables the analyst to check and compare historical cases with the current one in order to make a better judgment.

As a result of multimedia technological developments in the past decade, it has now become possible to store and process the digitized forms of images, voice and graphics of the output from the tested IC devices. Coupling this method with an object-oriented approach, we have developed a system for storing, processing and retrieving multimedia information to support IC failure analysis. This system is called MultiICFA. The objectives for the development of MultiICFA are 2-fold:

- to provide an environment for failure analysts to manage different forms of test output of the faulty IC

device—examples of these include video, image, graphical plot, photograph and text description;
- to develop an environment which is practical, feasible and cost-effective to implement.

Currently, multimedia technology with an object-oriented approach has been applied to many different areas. Dimitroff and Chang describe a prototyping system to automate patient diagnosis using an object-oriented approach [5]. Woelk *et al.* describe an object-oriented approach to multimedia document databases [18]. Meghini *et al.* discuss a multimedia document modelling and retrieval system called Multos [11]. Williamson and Stucky describe a geographical information system using an object-oriented approach [17].

Other multimedia applications have also emerged in areas such as patents and trademarks (registered trademarks and patents), museums (manuscripts, rare documents, paintings, statues, and furniture), tourism (tourist spots), police (suspects and criminals), and fashion technology (dress design and patterns) [10, 3, 16].

The objective of this paper is to describe the development of a multimedia information system for failure analysis using an object-oriented approach. The remainder of this paper is organised as follows. Section 2 reviews the characteristics of failure analysis. Section 3 discusses the functional specifications and architectural requirements for MultiICFA. Section 4 describes the architecture of the MultiICFA which is under design and development. Section 5 describes an object-oriented approach to the development of MultiICFA. In Section 6, the user interface design of MultiICFA is given. Section 7 describes the current status of the prototype. Section 8 discusses the future directions for MultiICFA. Finally, conclusions are given in Section 9.

3. FAILURE ANALYSIS FOR IC DEVICES

The objectives of failure analysis as stated in Burgess [2] are as follows:

- Determine cause of failure
- Identify symptoms
- Corrective action options
- Establish relevancy
- Reduce product failure rate
- Improve product performance
- Provide useful feedback
- Provide documentation record
- Develop failure history

As seen from the above, a speedier and more accurate analysis often gives an IC manufacturer an edge over its competitors. There are many approaches and procedures to determine the cause of an IC device failure. Each IC manufacturer normally develops its own checklist of tests. Figure 1 shows one possible failure analysis procedure. After receiving a request, the failure analyst carries out a series of tests as shown in Figure 1. The figure shows a typical test procedure. Variations to this procedure would be expected depending on the fault encountered. The meaning of each test and the corresponding output data are described as follows:

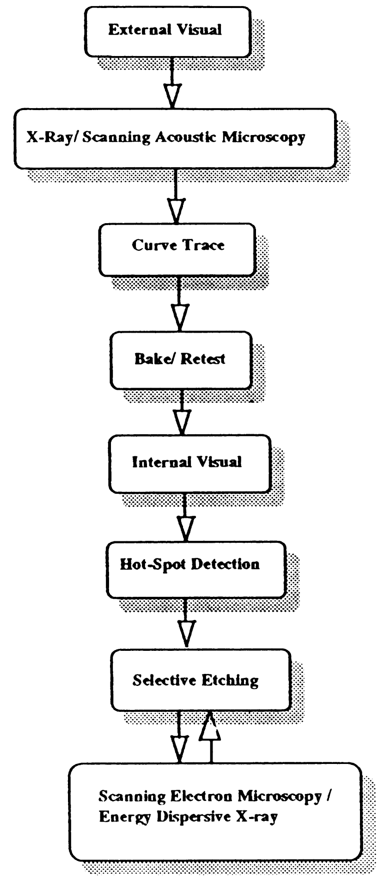

FIGURE 1. An example of a failure analysis procedure.

- *External Visual* – checks for any external physical anomaly as compared to a good unit. Two types of checking exist: package integrity (e.g. cracks, chippage, damage leads) and markings on the device (e.g. contamination). Output of this process may include photographs, video signals and text descriptions.
- *X-Ray* – checks the integrity of bond wires, die-attach, mold compound voids, lead frame, etc. Output of this process may include photographs, films, video signals and text descriptions.
- *Scanning Acoustic Microscopy or SAM* – checks for package problems such as voids, delaminations, cracks or any other discontinuities. Output of this process may include photographs, video signals and text descriptions.
- *Curve Trace* – checks the electrical characteristics of the IC pins. Typical information that can be derived from a curve trace includes open circuit, short circuit, leakage, and intermittency. Output of this process may include photographs, graphical plots, and text descriptions.
- *Bake* – units are basked and subjected to electrical test to check for ionic contamination or instabilities.

Output of this process may include graphical plots and text descriptions.

- *Internal Visual* – is performed with the units being decapsulated. Die will be first inspected at low power magnification (30X) and the following defect modes will then be inspected: wire loop, die, type of die, bonding problems, foreign materials. High power magnification is used to look for the following defects on the die: passivation, metallization, polysilicon, mask identity, foreign material, crack or chippage on die, and corrosion or contamination. Output of this process may include photographs, video signals and text descriptions.
- *Hot-Spot Detection* – is performed by using liquid crystal to localise a 'hot spot' on die. The hot spot observed will narrow down the area of the circuit where analysis will be carried out. Output of this process may include photographs, video signals and text descriptions.
- *Selective Etching* – die deprocessed layer by layer using plasma and chemical etching techniques. The die will be examined under an optical microscope for defects. Output of this process may include photographs, video signals and text descriptions.
- *Scanning Electron Microscopy or SEM* – is used for detection and identification of photoresist residues, metallization cracks, metal step coverage, pinholes or any other defects which are difficult to detect using an optical microscope due to insufficient resolution or depth of field. Output of this process may include photographs, video signals and text descriptions.
- *Energy Dispersive X-Ray* – is used to determine the elemental composition quantitatively and qualitatively. Output of this process may include photographs, graphical plots, video signals and text descriptions.

After each test, a large volume of data for each tested IC device is produced. Moreover, these outputs come in different forms and formats which need to be captured, stored and made accessible. The analyst is inundated with data after a series of tests for a particular IC device.

3. FUNCTIONAL SPECIFICATIONS AND ARCHITECTURAL REQUIREMENTS

This section presents the specifications of MultiICFA, an event-driven system. The development is based on a multimedia database system with a friendly user interface. In MultiICFA, test results in different forms and formats will be stored in the server computer. After all required tests are performed, the analyst can retrieve and view different test output for an IC device, either by querying on the search items (e.g. device name, failure mode, and so on) or through other appropriate commands.

Before describing the system architecture, it is necessary to review the major goals behind the development of MultiICFA. From a functional point of view, the architectural requirements for MultiICFA can be divided into three areas:

- *Networking Support:* The system, must be able to transmit and receive multimedia objects over a computer network.
- *Friendly User Interface:* The system should be able to give the analyst easy access to different types of test output of faulty IC devices. The system should therefore support an interactive retrieval system with an event-driven query (clicking mouse, selecting menus and so on). The analyst should also be allowed to express a query consisting of attributes as well as keywords for searching past test cases.
- *Object-Oriented Multimedia Support:* The system should be able to support various information objects such as text, image, graphics and audio. Thus the system should be able to display, manipulate, store and retrieve different data structures. An object-oriented data model can be used to model every conceptual entity by using a single modelling concept: the object. An object-oriented program encapsulates data with procedures that act upon the data. An object is a package of data and a set of operations. Objects cleanly separate external specification from internal implementation. Operations defined under an object capture its behaviour and allow it to manipulate data in an abstract way without referring to the implementation details. All interactions with an object occur through messages that request it to execute one of its operations. In addition, objects are grouped to facilitate reuse of similar code. Abstraction mechanisms such as aggregation and generalization can be used to represent relationships among objects, and among object collections. The rationale of an overall system design based on object-oriented modelling is that a straight forward integration with object-oriented programs is possible as both data models and programs use a common metaphor and the same design notation to provide a unified system specification. Moreover, the object-oriented approach can be used to construct efficient management and operation methods.

In view of the above architectural requirements, it is felt that an object-oriented approach is most appropriate for the construction of MultiICFA. Many of the test output have similar attributes: video, film, image, graphical plot and text description. They also have similar operations defined upon them: display, print and store.

4. SYSTEM ARCHITECTURE

The overall system architecture of MultiICFA is shown in Figure 2. This is the proposal of the intended system upon which prototypes are currently being built. The system is to be based on a client-server distributed system [15]. Within the failure analysis department of

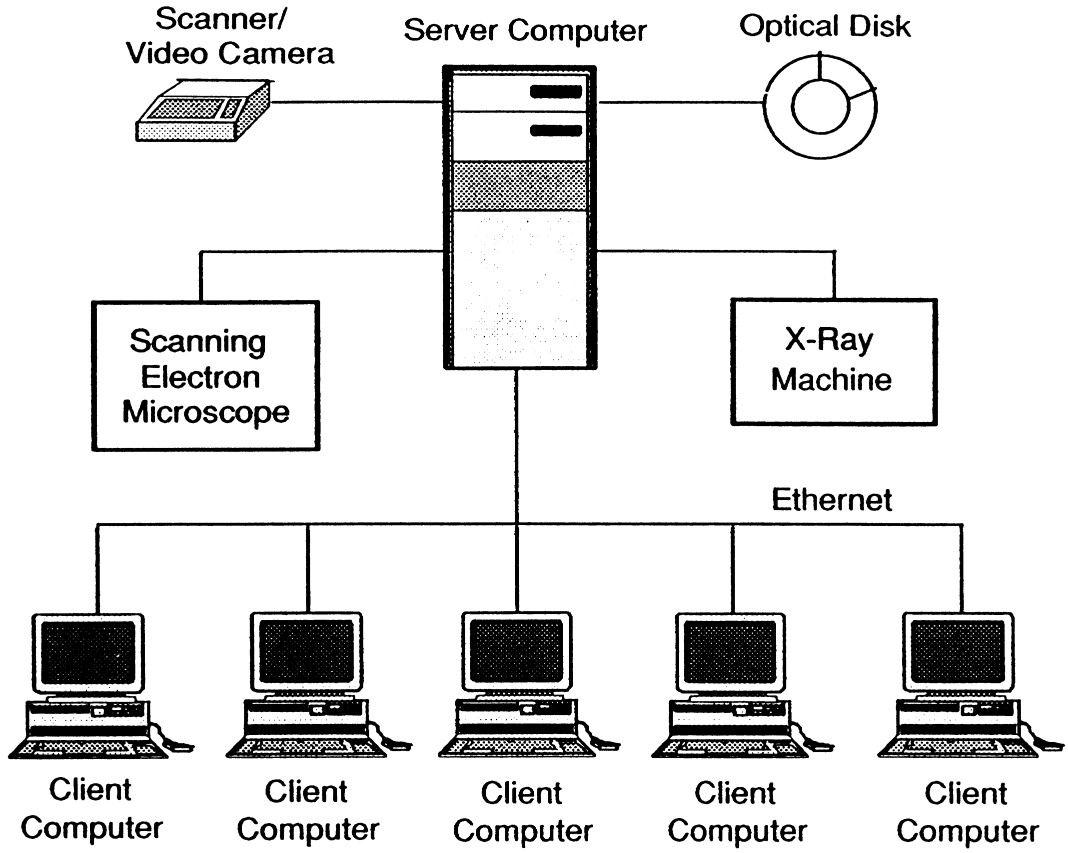

FIGURE 2. MultiICFA architecture.

the IC manufacturer, a set of personal computers will be interconnected to a server by a local area network. These personal computers are to be scattered around different test points of the department. An optical disk system will be attached directly to the server. Special purpose testing machines (X-ray, Scanning Acoustic Microscope, etc.) may be connected directly to the server. For test equipment which is not directly connected to the server, the digitized image of the output data can be captured through a digital scanner or a video camera which is attached directly to the server. These images will be stored in the optical disk for later retrieval. After all the required tests have been performed, the failure analyst can interact with the system to retrieve different test results using the user interface provided. It is also possible for the failure analyst to retrieve the history of a tested IC device for reference from the server.

Images in digitized form take up a great deal of storage space. In MultiICFA, an image which has been digitized may be stored on any magnetic storage medium such as a hard disk. But the vast space required for image storage renders this option impractical. Data compression techniques can be used but this imposes overheads on the system. In addition, it will affect the quality of the image stored and retrieved. Optical disk technology offers the large storage capacity required. Digital optical disks can hold image, text and audio on a single medium in digital form that enables manipulation of data. Optical disks offer just the right solution for MultiICFA by providing long-life, low-cost, high volume data storage and retrieval [6].

5. OBJECT-ORIENTED APPROACH

In order to meet the design objectives of MultiICFA, stated in Section 3, an object-oriented approach is adopted both for data modelling and programming. The design of an object-oriented data model and program is closely coupled since operations to manipulate the objects have to be considered and implemented during data model construction. The object-oriented data model is very similar to the conceptual model of a conventional relational system except that operations and methods describing the behaviour of each object are included within the data model [4, 1, 7, 13].

For the failure analysis application, an object-oriented data model is constructed using the Object Modelling Technique (OMT) [12]. Figure 3 shows an object class hierarchy of MultiICFA. The boxes in Figure 3 denote object classes. Each object has a set of attributes and operations. An object talks to the rest of the world via messages. Device, Requester, Report and Failure Testing Record are examples of object classes. The details of each class are described as follows:

- The Requester class contains information such as requester's name, department, telephone number and electronic mail address. The name of the analyst who is going to perform the failure analysis is also contained in the Requester class.

- The Device class contains data such as the IC failure mode, IC device name, package type, quantity received for analysis, date code and request type. It also contains information about when the IC device failed: during production or manufacturing.
- The Report class contains the conclusion of the failure analysis which is entered by the failure analyst as text description data at the end of the analysis.

A generalization represents the 'is-a' relationship, by which classes can be organized into taxonomies. A class defined as a generalization is called a superclass, and subclasses inherit attributes and operations from their respective superclasses. Generalization may have an arbitrary number of levels. This means that reusability and extendibility in programming languages are maintained. The coded programs can be reused or modified and extended by inheriting the common parts and adding new features. The heavy triangle in Figure 3 symbolizes generalization. The generalization structures identified are as follows:

- Each Failure Testing Record object can belong to categories such as Class-1, Class-2, and so on. For higher generalization, the failure testing record is the superclass; the categories such as Class-1 are subclasses. The superclass stores data pertaining to the test in general. The subclasses store data particular to each type of test. For example, Class-1 could contain video output whereas Class-2 could refer to graphical plot data. Similarly, for the lower generalization, Class-1 is the superclass while External Visual, X-Ray, SAM, Internal Visual, Hot-Spot and SEM are subclasses. These subclasses are the individual tests which produce the respective categories of data.

Attributes are inherited from the top level down in the generalization structure. For example, each External Visual consists of a text description, video and photographs.

An association relates two or more independent objects. A number of association relationships are identified.

- A one-to-one relationship is identified between Device class and Report class. In other words, one report is generated for each tested faulty IC device.
- Two one-to-many relationships are identified. One is between Requester class and Device class. This means that one requester can request for one or more faulty IC devices to be tested. The other one is between Device class and Failure Testing Record class. This refers to one or more tests that can be carried out for one faulty IC device. Thus, a number of failure testing records will be generated for each faulty IC device.

An object structure is defined as a combination of the data structure (attributes) and operations attached to the object. The implementation of the data structure and operations are hidden from the analyst. For each Failure Testing Record, a set of operations can be defined. Operations for adding new records to the MultiICFA and for displaying them are included. Operations for

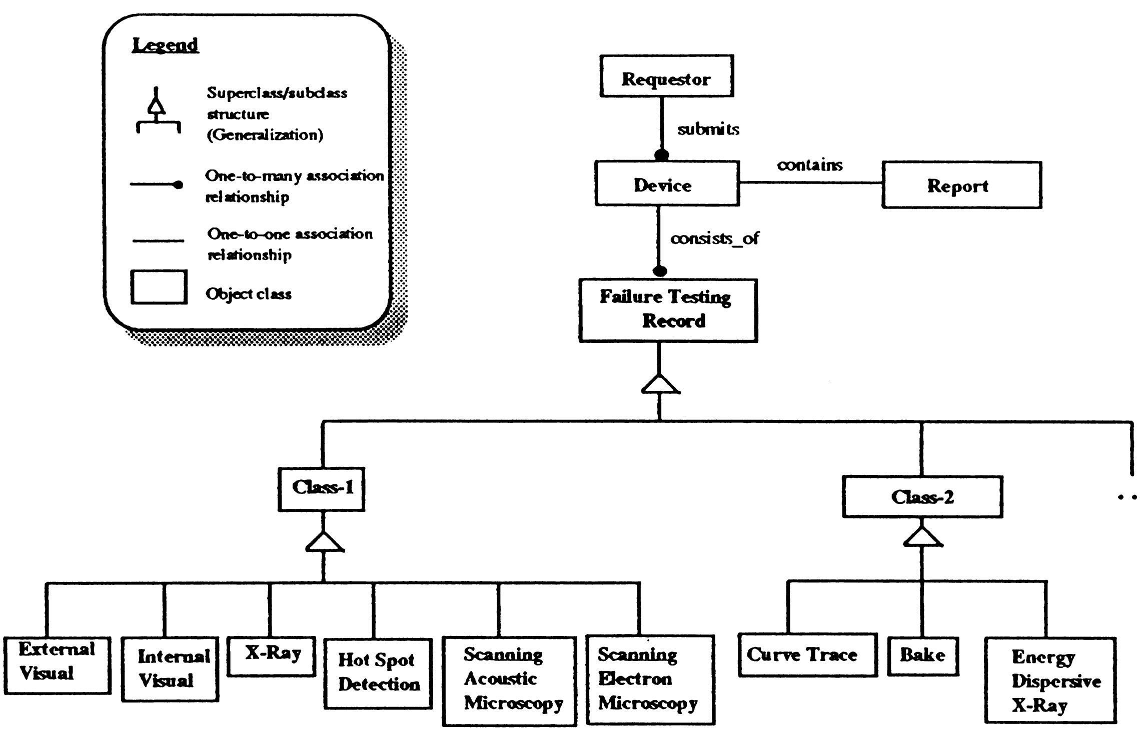

FIGURE 3. Object class hierarchy of multiICFA.

deleting and updating existing records are also available for each Failure Testing Record. Through the inheritance property, all these operations are inherited by the subclasses of the Failure Testing Record. Thus all these operations are made available for execution in all subclasses. Similarly, operations for saving, adding, updating and deleting are also defined for Requester, Device and Report object classes. The type polymorphism concept of object-oriented programming is used. It refers to the ability to invoke the same routine on objects of different classes. For example, the generic operation or method, *Display*, can be invoked on all the object classes shown in Figure 3. Each routine is tightly bound both to the object class it acts on and to a generic operation name through which it can be called. This means that when the operations are invoked upon an object class during run-time, the object class having the appropriate operations will be executed.

For each faulty IC device, all related objects can be displayed as a group of options on the screen. There are options pertaining to the information about the requester, the report conclusion, and a set of tests which have been carried out. For example, if the user clicks-on the conclusion selection on the screen, another window will pop up which contains a detailed report on the tested faulty IC device; if the user clicks-on the X-ray test option, then another window containing photo, film, video and text description sub-menu will pop up. The user can then select which test output to view by clicking the corresponding option on the screen.

6. THE USER INTERFACE

The design of the user interface is one of the most important issues of MultiICFA. Due to the job requirement, the failure analyst needs to identify the cause of failure as early as possible. The user interface provided by the system should enable the information to be retrieved and manipulated by the analyst in a user-friendly fashion. In addition, there are three other goals [14, 9]:

- To be consistent and uniform;
- to be appropriate for expert and novice analysts; and
- to be flexible to the needs of individual analysts.

MultiICFA provides different ways of accessing the stored information in a user-friendly environment. The user interface supports interaction between the analyst and the system. It also acts as a filter between the input devices and the stored database of tested faulty IC device records. Three types of interaction are available in MultiICFA:

- *Menu Driven Interaction* – operations (such as display, browse, insert, etc.) are presented in a pop-up menu. The analyst can then select the corresponding operation with a mouse.
- *Object Driven Interaction* – options which represent different tests performed on a faulty IC device (such as X-ray, SEM, Curve Trace) are displayed. The analyst may select a displayed option to display the output data.
- *Query Driven Interaction* – this mode of interaction occurs during record retrieval operation. When a Display operation is invoked, a new window is generated. The analyst may then input a query with some specified conditions. The system will retrieve and display the tested IC device records which satisfy the specified conditions.

The screen is divided into four different areas as shown in Figure 4. Proceeding from top to bottom, we have:

- *Menu area* – used for command selection.
- *Display area* – used for options and record data display.
- *Message area* – used for system responses, error messages, help messages and other information resulting from user interactions.
- *Command area* – keyboard input is supported through this area. A skilled analyst may use this area to input a command in order to speed up interaction with the system.

In a multimedia environment, queries are more complex than those in a traditional relational database system. Multimedia record retrieval should be based on a Boolean query language which allows one to specify restrictions on attributes and on the free text. Record retrieval in MultiICFA is supported by three different operations of the Display command in the menu:

1. The analyst can formulate a query by selecting the Query option. A query window is displayed as shown in Figure 5. The window is divided into two areas. The upper left portion displays the selections which can be made by the analyst. It includes device information, requester information and report information.

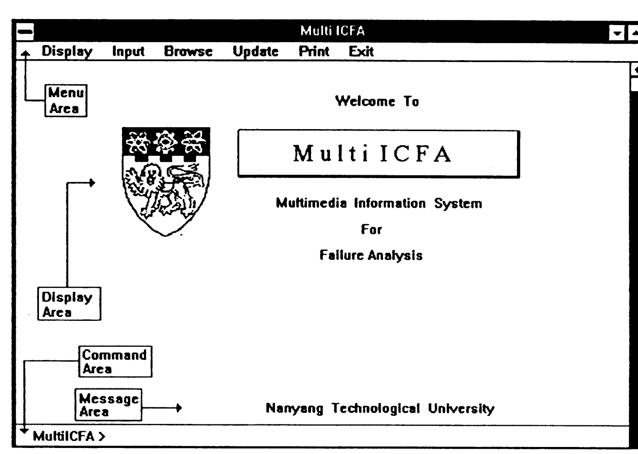

FIGURE 4. Screen layout.

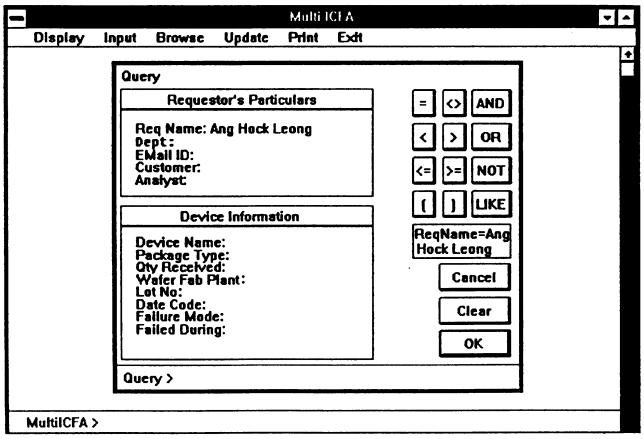

FIGURE 5. Record retrieval by query.

The right portion displays selection conditions (AND, OR, NOT, LIKE, etc.) from which any restrictions on retrieval can be formed. Attributes of each object are also displayed. The analyst simply clicks-on any attributes and the selection conditions to formulate a query. In addition, the user can also specify keywords (using the operator LIKE) for free text retrieval on the textual description of each tested faulty IC device. The lower part of the window allows the analyst to enter any query commands. This region allows expert analyst to type in queries directly to speed up accessing.

2. The analyst can invoke any past executed query or query history, and reformulate a new query based on the past query by selecting History option. A list of past queries are displayed, and the analyst can inspect and choose one query to work on. This option is very useful in giving the analyst an idea of which queries have been issued and to speed up query formulation.
3. After the query is formulated, the analyst can execute and display those qualifying records by selecting the Execute option.

Each record is displayed as shown in Figure 6. Device information, requester information and a conclusion will be displayed as options with summary information. The analyst can click-on the required option to retrieve the details. Test output will also be displayed as a set of options which represent External Visual, X-ray, Scanning Acoustic Microscopy, etc. Since it may take time to display a detailed image output on the screen, the analyst can first browse through the device information, requester information and conclusion before viewing any test image output. Once the desired record is located, the option can then be clicked-on to display the output. By doing this, the analyst can skip over records and image output which are not relevant. Figure 7 shows an example of a Scanning Electron Microscope test output.

The user interface also provides an Input command for the analyst to create new records, a Browse command to browse through the faulty IC device database, an Update command to change or update an existing data record and a Print command to print a report of the faulty IC device.

7. CURRENT STATUS

Two separate prototype systems of MultiICFA are now under development at the Nanyang Technological University. This is being done using two different database products. The first is a multimedia relational database system running on a personal computer. The server consists of an IBM PC compatible. Client personal computers can also be any IBM PC compatibles connected with a commercial scanner. Client personal computers communicate with the server via Ethernet. The user interface is written in a fourth generation programming language. The communication software of the client personal computer and the server is written in C. The choice of a cheap, low-end personal computer is intended to make the solution affordable to typical IC failure analysis departments. Obviously, performance is a major issue. Our prototype is based on the following configuration:

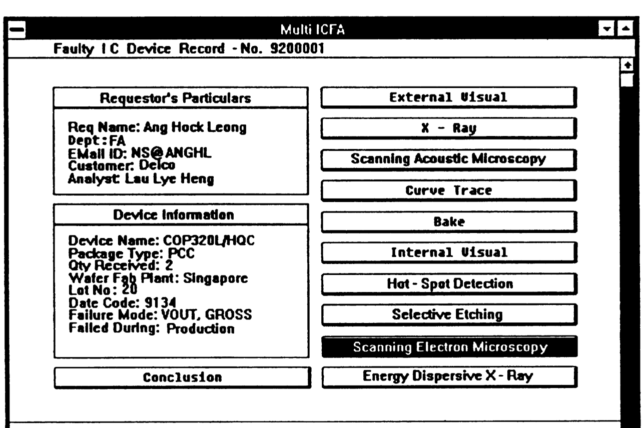

FIGURE 6. A faulty IC device record.

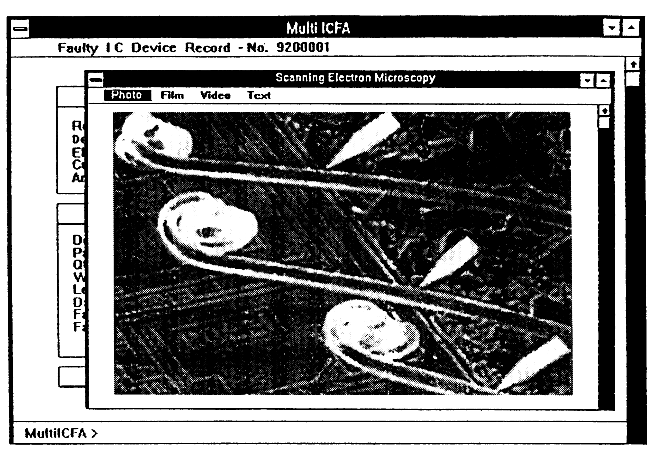

FIGURE 7. An example of scanning electron microscopy.

Processor: Intel 80286 (12 MHz)
Math Co-processor: Intel 80287
Memory: 4 Mbytes
Display: VGA
Hard Disk: 180 Mbytes
Average seek time of 22.26 ms
Data transfer rate of 576 Kbytes/sec.

Photographs are saved as 8-bit TIFF files. Each image file requires about 75 Kbytes. At present, videos are not stored, but testing is being carried out live with a video camera. Measurements of the response time required to retrieve an image file from the hard disk and to display the image onto the screen show that an average of 4.1 seconds was needed. This is an acceptable response time considering the relatively slow system used.

The second prototype is being built on workstations using an object-oriented database product. A similar configuration to that mentioned above is used, except that both the server and the terminals are workstations. The object-oriented database provides a closer mapping of the model shown in Figure 3. The database provides a rich set of development tools to construct the underlying data classes, user interface and communications management. The system is still in the process of being developed and thus, no figures on storage and other performance indicators are available.

8. FUTURE DIRECTIONS

Further ideas for incorporation to MultiICFA in the future include:

- *Graphics.* To model the faulty IC device in order to present various views and perspectives. The analyst is provided with functions to rotate the three dimensional model of the faulty device at different angles.
- *Pattern recognition.* To assist in the recognition of classes of failure with similar device. The analyst is allowed to query the system by image content for record retrieval.
- *Overlaying information.* In transmitting documents over the network, it would be useful to have the ability to mark on documents. Examples of this include the use of handwriting or encircling of specific parts within an image or document to convey some message to the receiver of the document.

9. CONCLUSIONS

This paper has described a multimedia information system, MultiICFA, for IC failure analysis. The system allows failure analysts to specify queries to retrieve IC device records from the multimedia database system through an event-driven user interface. The paper has reviewed an object-oriented approach towards the development of the prototype system. The development of the MultiICFA for storing, processing, and retrieving multimedia information shows considerable potential in the area of failure analysis. The benefits of such technology are as follows:

- All faulty IC device records are stored on a server, so that from a central location, a pool of information on failure history can be made available to analysts through the local area network.
- Storage of digitized versions of images, photographs, and graphical plots is able to stem the natural decay of the original. References access the stored digitized versions without touching the originals.
- Analysts are able to produce failure analysis report of the faulty IC device upon request.
- The time to perform each analysis may be reduced, since analysts are guided by the system to locate the desired information instead of looking up a pile of paper filing records.
- This is one of the safest methods of data storage, since the digitized versions of images, graphical plots and photographs will be stored on optical disks which have a fairly long life span.

Acknowledgments

We wish to thank our colleague Hiok-Chye Quek for his helpful comments and suggestions.

REFERENCES

[1] G. Booch, *Object-Oriented Design with Applications.* Benjamin Cummins Pub. (1991).
[2] D. L. Burgess, *Failure and Yield Analysis Handbook*, Technology Associates, Portola Valley, California (1987).
[3] A. Chaudhry and A. Roy, Art records treasury system: Picture retrieval through image databases. *Online Information 90, 14th International Online Information Meeting Proceedings* (1990) pp. 82–87.
[4] P. Coad and E. Yourdon, *Object-Oriented Analysis.* Yourdon Press (1990).
[5] D. C. Dimitroff and I. F. Chang, An object oriented approach to automating patient medical records, *IEEE 14th Annual International Computer Software and Application Software Conference* (1990).
[6] N. L. Eaton, L. B. Macdonald and M. R. Saule, *CD-ROM and other Optical Information Systems*, Oryx Press (1989).
[7] D. W. Embley, B. D. Kurtz and S. N. Woodfield, *Object-Oriented Systems Analysis: A Model-Driven Approach*, Prentice-Hall (1992).
[8] R. Gupta and E. Horowitz (eds), *Object-Oriented Databases with Applications to CASE Networks, and VLSI CAD*, Prentice Hall (1991).
[9] Hartson and Hix, Human-computer interface: Concepts and system for its management, *ACM Computing Surveys*, **21**(1) (1989).
[10] K. A. Jordan and A. J. Zawilski, Specification of a rapid prototyping capability for the automated patent system. *IEEE 14th Annual International Computer Software and Application Software Conference* (1990) pp. 76–81.
[11] C. Meghini, F. Rabitti and C. Thanos, Conceptual modeling of multimedia documents. *IEEE Computer* (1991) pp. 23–32.
[12] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and

W. Lorensen, *Object-Oriented Modeling and Design*, Prentice-Hall (1991).

[13] S. Shlaer and S. J. Mellor, *Object-Oriented Systems Analysis: Modelling the World in Data*, Yourdon Press (1988).

[14] B. Shneiderman, *Designing User Interface: Strategies for Effective Human-Computer Interface*, Addison-Wesley (1987).

[15] L. Svobodova, File servers for network-based distributed systems. *ACM Computing Surveys*, **16**(4) (1984).

[16] P. Wentz, Computerization in museums: Databases to image bases. *Online Information 89, 13th International Online Information Meeting Proceedings* (1989).

[17] R. Williamson and J. Stucky, An object-oriented geographical information system. In *Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD*, Prentice Hall (1991) pp. 296–312.

[18] D. Woelk, W. Kim and W. Luther, An object-oriented approach to multimedia databases. *International Proceedings of the ACM SIGMOD Conference in the Management of Data* (1986) pp. 592–606.