

# AMI: The Automated Maintenance Instruction System<sup>1</sup>

James P. Buckley  
Frank W. Moore  
University of Dayton  
Computer Science Department  
300 College Park Drive  
Dayton, Ohio 45469-2160  
(buckley@cps.udayton.edu)

## Abstract

Various organizations, including the government, are required to produce documentation that facilitates the maintenance of a variety of equipment, such as airplane motors. Maintenance manuals are expensive to produce and update, and are generally not easily transportable. Additionally, the end user/maintainer has little if any communication with the designer/manufacturer during the development process. The Automated Maintenance Instruction (AMI) system described in this paper improves upon existing techniques by employing object-oriented concepts and planning methodologies. The AMI system resolves all of the aforementioned problems as well as providing additional features, such as interactive access to mission-critical maintenance information available in multimedia format on the World Wide Web (WWW).

## Introduction

Government, military, and industry organizations are required to maintain complex systems operating in diverse environments at various locations around the globe. Experience has forced these organizations to adopt procedures for developing manuals that can be understood and used by relatively inexperienced maintenance personnel. However, such documentation is expensive to produce. Additionally, since existing systems are constantly being modified and upgraded, it is virtually impossible to maintain accurate maintenance manuals after system deployment.

This "traditional method" of producing maintenance documents also excludes the needs of maintenance personnel during the design and manufacturing phases. Such input, in many cases, would result in a better and more maintainable end product.

Interactive electronic technical manuals (IETMs) use multimedia (including virtual reality, audio/visual displays, graphics, and text) to provide a welcome alternative to the traditional paper-based Technical

Orders. They provide greater ease of access as well as more reliable updating (Ventura 1988). However, without access to a centralized knowledge base, these IETMs can also become outmoded after deployment.

The Automated Maintenance Instruction (AMI) system described in this paper improves upon existing techniques by employing object-oriented concepts and planning methodologies (Hendler, Tatae, and Drummand 1990, Rivera 1996). AMI provides reliable, interactive access to mission-critical maintenance information available in multimedia format on the World Wide Web (WWW). This technology makes the distribution and maintenance of paper documentation obsolete, and eliminates the necessity of having to deploy physical copies of modified IETMs at numerous geographical locations.

## Overview of AMI Design

Figure 1 provides an overview of the AMI system.

The maintenance procedures must be fully specified by the expert (designer/manufacturer) to suit the needs of maintenance personnel. Maintenance needs are specified by maintenance personnel for specific types of components/systems and must be addressed sufficiently by designer/manufacturer during product development. The AMI interface facilitates this specification.

As a result, the needs of maintenance are considered early on in the design and/or manufacturing life cycle

Given a primary (root) component and all subcomponents, a hierarchical tree is constructed. From a designer point of view, the component tree is built bottom-up starting with atomic subcomponents. The maintainer can enter the tree at any level, including the root. The maintainer traverses the tree until one or more atomic subcomponents are reached.

---

<sup>1</sup> Copyright © 1998, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

The hierarchical tree is maintained as a set of objects. Each object has the generic format as specified in Figure 2.

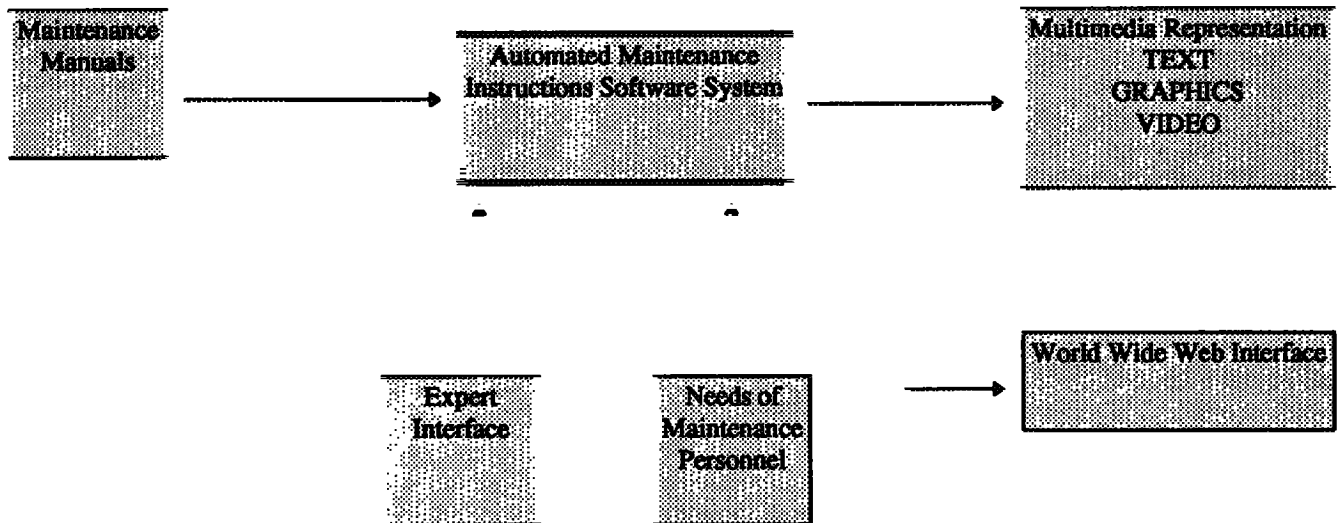


Figure 1. Overview of the AMI System.

```

Object Name: <o_name>
Object Identifier: <oid>
Hard Relationships:
<hrel_1>
<hrel_2>
<hrel_3>
<...>

User-Defined Relationships:
<udrel_1>
<udrel_2>
<udrel_3>
<...>

Procedures:
<proc_1>
<proc_2>
<proc_3>
<...>
  
```

Figure 2. Format of AMI Objects.

The object name is used to identify the specific component to the outside (user) level. The unique identifier of each object is specified by the object identifier. Hard Relationships are those that hold true for all instantiations of the object. For example, an airplane wing will always have a relationship with at least one specific airplane type. However, there are many relationships that are unique for a specific instantiation of an object. These are specified in the User-Defined Relationships portion of an object. They provide for both flexibility of objects within a system and reusability of objects by other related systems. Lastly, each object has an associated set of maintenance procedures. The path to these procedure steps is located within each object.

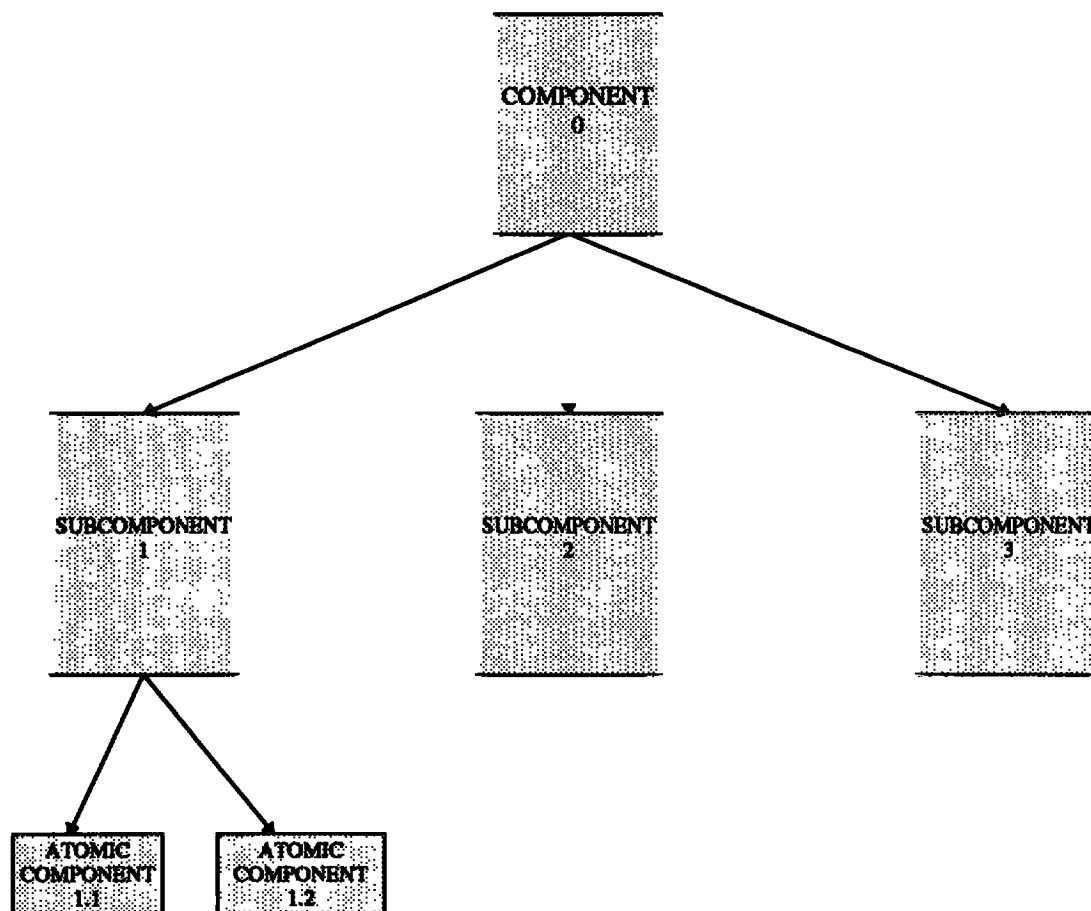
### AMI Object Hierarchy

Object oriented concepts have been discussed extensively in the literature (Booch 1991). In the AMI system, a hierarchy of objects is constructed as follows: Each

component is divided into objects, where each object represents a "subcomponent-of" relationship. We proceed in this manner until atomic components (leaf nodes) are reached. Each atomic component is then defined fully and all associated maintenance procedures are attached. When more than one atomic object accesses the same maintenance procedure, the link to the procedure is moved from each object and represented only once in the parent. This approach guarantees that our tree is minimal in terms of linking constructs.

The user can then use the AMI system to specify a class hierarchy of object types and associated maintenance procedures. When an object is instantiated, it inherits all procedures associated with its class. By use of an expert interface, the user can arbitrarily tailor procedures as needed.

In terms of objects, the hierarchy would be represented with the objects and subtypes as shown in Figure 3.



**Figure 3. Hierarchical Specification of AMI objects.**

Procedures must be specified for each object in the AMI knowledge base. Each of these procedures references a step or series of steps to accomplish the corresponding maintenance procedure for the specific component the object represents, as shown in Figure 4. Each procedure is represented as a linked list of tokens identifying

specific maintenance steps. Therefore, each token is a link to an atomic description element that describes that particular step. The description elements themselves are maintained in a non-redundant fashion and it is probable that more than one step in separate objects may link to the same description element.

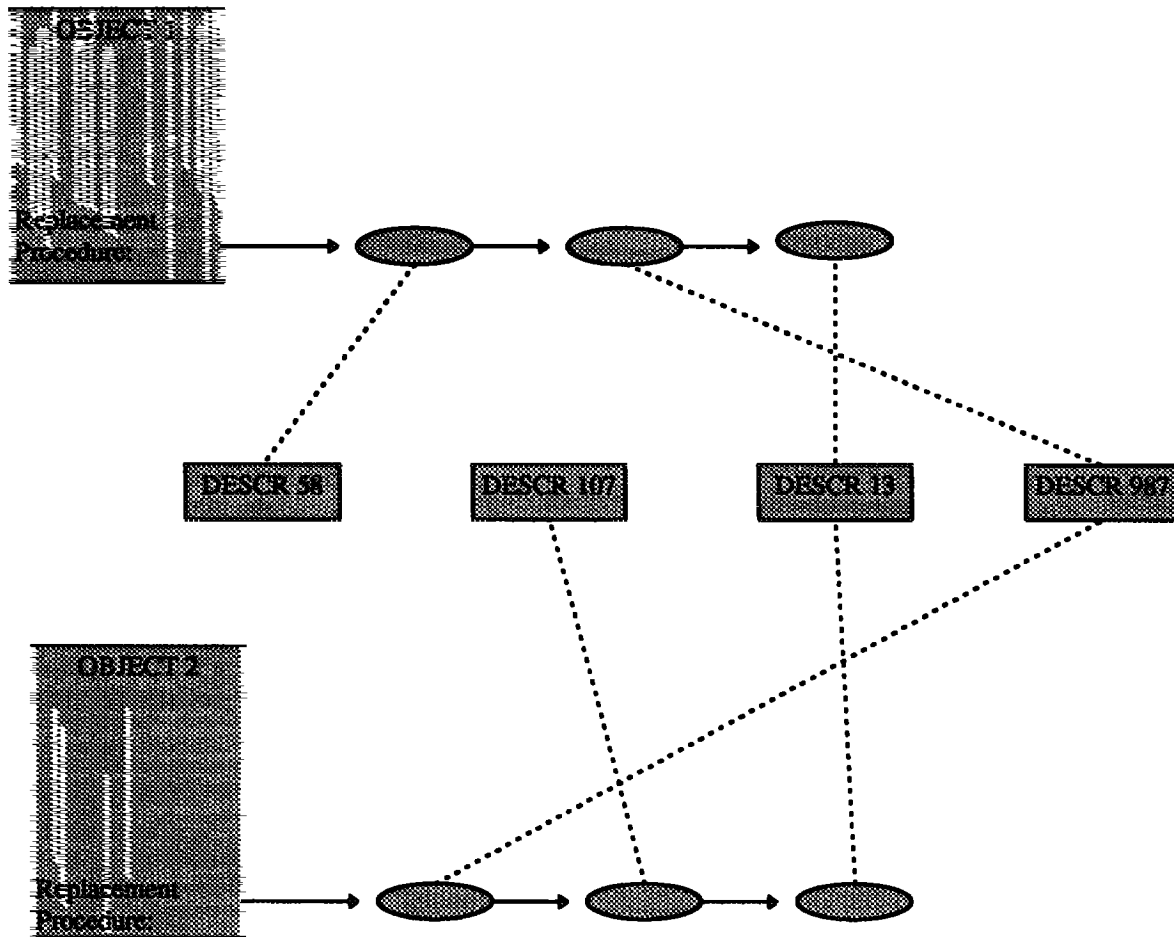


Figure 4. Decomposition of Maintenance Procedures as Atomic Steps.

The AMI system allows domain experts to develop a knowledge base specifying the properties, attributes, and sub-components of new and existing systems and subsystems. AMI's class-based hierarchical decomposition may be repeated to an appropriate level of detail for each deployed system. Centralized control of AMI content allows domain experts to rapidly update the AMI knowledge base in response to incoming Change Requests and Problems Reports. Using hand-held computers, inexpensive satellite communication technology, and well-established data encryption and

coding mechanisms, fielded maintenance personnel are afforded immediate worldwide access to the most up-to-date information available.

## Conclusions and Future Directions

The traditional "paper-based" approach to producing maintenance documents has many problems, including costly production and updates, technical manual distribution, and lack of maintenance feedback during design and development. The AMI system as described

in this paper provides a number of distinct advantages over previous approaches. The end user is actively involved in the design and manufacturing phases. The resultant documentation is electronically represented, reducing cost and providing for more efficient and reliable updates. Since the maintenance documentation is stored in a central location and can be accessed from any location geographically, it is more portable. The object hierarchy approach leads to potential reusability for future documentation projects. The AMI system's multimedia approach simultaneously improves the quality of maintenance instructions while reducing the cost of maintenance procedures and significantly improving reliability.

Some areas of future research are:

- How is the security of information on web pages insured?
- Since we cannot automatically extract all relevant data in a format that is useful to maintenance personnel, sources such as CAD data will be used to retrieve information automatically.
- In terms of non-automated input, an expert system should be implemented for extracting additional necessary maintenance information from experts (designers/manufacturers).
- Also, a near-natural-language user interface should be implemented for retrieving information stored in the knowledge base.
- In order to sufficiently maintain the multi-level capabilities of AMI, information should be

represented in a format that can be retrieved easily by novice maintenance personnel, since the data representation does not in itself facilitate access to maintenance procedures.

## Acknowledgments

We would like to thank Dr. Bapu at the University of Dayton Research Institute for his invaluable contributions to the work described in this paper.

## References

- Booch, G. 1991. *Object-Oriented Design With Applications*. Addison-Wesley.
- Hendler, J., Tate, A. and Drummand, M. Summer 1990. AI Planning Systems and Techniques. *AI Magazine* Vol 11 No 2.
- Rivera, G. 1996. Human Factors Issues in Interactive Electronic Technical Manual for Aircraft Maintenance. <http://members.aol.com/geo13/ietm.htm>.
- Ventura, C. 1988. Why Switch From Paper to Electronic Manuals? In Proceedings of the ACM Conference on Document Processing Systems, Santa Fe.