

Performance-based Documentation

Malcolm Graham
WriteDoc Inc.
malcolm@writedoc.com

This paper presents a method for designing product documentation that relates information to acceptable user performance in acquiring specific skills and knowledge. The method involves selecting, organizing, and presenting all information using a performance-based development model. The use of the model is illustrated with examples from actual projects.

Information purposes

Before selecting and designing document content, it is important to:

- define the intended purposes of the content
- identify how the intended purposes will be achieved
- know how the success of the intended purposes will be verified

The main purpose of product documentation is to communicate specific information to users in such a way that they will be able to demonstrate the acquisition of specific skills and knowledge. There are six main generic sub-purposes; they are, to:

- **Buy** the product
- **Plan** for product installation and use
- **Train** how to use product features
- **Install**, test, and commission the product
- **Operate** the product
- **Maintain** the product

Identifying and verifying purposes is facilitated by the use of information objectives.

Information objectives

In order to acquire specific skills and knowledge, the required performance, conditions, and criterion of acceptable user performance must be identified. This is achieved by the use of behavioral objectives. A behavioral objective is a description of a performance you want users to be able to demonstrate before you consider them competent. Each objective describes an intended result of the information rather than the information structure. A clear statement of objectives also provides guidance for the selection of materials, methods, and the means for assessing information effectiveness.

Objectives have been used for many years during the preparation of instructional materials; this paper simply shows how the same approach can be used for preparing product documentation. For general guidelines on how to define effective objectives see Mager, 1973 and Mager, 1975.

Objectives are used to relate product information and its presentation to tasks that users need to perform. The overall behavioral objective of product information is that users will be able to use the product efficiently and effectively; in order to achieve this, users will have to be motivated to learn new behaviors. To define useful objectives, certain adult learning behaviors need to be considered.

Adult learning and objectives

Most adults resist learning anything new and they avoid reading. Adults try to use new systems without training or reading the appropriate documents (even when they are available and well designed and written). In new learning situations, adults often resort to guessing how a new system will behave. This behavior is reinforced by the fact that most adults have been conditioned to have a production bias where throughput is paramount; they are expected to use new products immediately and time spent thinking and reading on the job are not generally viewed as productive behaviors. This conditioning reduces any motivation to spend time learning about anything new. Adults must rely on their existing knowledge for interpreting and assimilating new information; this reliance can be very useful but only if the difference between existing and new knowledge is clearly distinguishable; otherwise, existing behaviors will be reinforced rather than new ones created.

These behaviors could adversely affect how quickly users will learn how to use the product. In order to minimize the possible adverse affects of these adult learning behaviors, users need:

- information presented in a way that minimizes reading and searching (for example, by use of visuals, lists, video, and context sensitivity)
- immediate feedback on task completion
- information presented in a way that links to their existing knowledge but which is clearly distinguishable from it

In order to address these needs, each behavioral objective needs to:

- define a small enough task that can be quickly assimilated so that users can receive immediate feedback on task completion
- be clearly presented so that users are aware of the criteria for successful task completion
- be part of a hierarchy of expected behaviors to provide a context for each user action (this is achieved by defining behavioral objectives for: the complete information suite; each deliverable; each chapter; and each module of information)
- use language that will be familiar to the intended users
- be phrased in a way that they can form the basis for conducting formative usability (performed on individual information modules while the document is being developed, see Kohn Kaminsky, 1992) and summative usability (performed on the complete document) to test for information product effectiveness

Project examples

Table 1 presents a number of composite examples from actual projects that illustrate the use of behavioral objectives.

Document type	Purpose	Behavioral objectives (users will be able to do the following:)
Introducing the product	Buy Plan Install Train Operate Maintain	<ul style="list-style-type: none"> • State the purpose of the product • List the main components of the product
Sales and Ordering Guide	Buy	<ul style="list-style-type: none"> • Describe the main product features and their value • Complete an ordering form
Planning Guide	Plan	<ul style="list-style-type: none"> • List the main components of the product • Draw a diagram that shows product relationships • Write a brief description of product • Draw a diagram of the product architecture • Locate and list the available information products • Complete a sample planning exercise
Installation Guide	Install	<ul style="list-style-type: none"> • Identify main hardware and software components • Locate and list the available information products • List the main installation tasks • List the installation equipment and tools required • Start-up the product • Verify correct installation
User Guide	Train Operate Maintain	<ul style="list-style-type: none"> • Locate and list the available information products • Identify main hardware and software components • Monitor product-related activity • Capture and organize data logs • Identify common alarms and failures • Execute troubleshooting procedures • Locate and repair or replace faulty components

Table 1: Product information purposes and objectives

Information categories

Identifying your users, the tasks they perform, defining behavioral objectives, and testing for the effectiveness of each information module is of little use if the right information for specific users can't be found when they need it. The structure of the information must be mapped to user tasks and information provided at the appropriate point during task performance. This can be achieved by categorizing the objectives defined to test for competent task performance and, where appropriate, providing context-sensitive information directly from the user interface.

I have identified four main generic, hierarchically-related, information categories:

- **users** (associates information with specific types of users)
- **purposes** (associates information with its intended use by users)
- **tasks** (associates information with the tasks that have to be performed by users)
- **primitives** (associates information with the type of information needed by users)

Each generic information category bears a hierarchical relationship to each other—**users** is the parent of **purposes**; **purposes** is the parent of **tasks**; and **tasks** is the parent of **primitives**.

The *users* category enables information to be associated with the information needs of users and could be used as the entry level for the other categories from either the product API (application programming interface) or the World-Wide Web. The *purposes* category enables information to be associated with the intended use of the information. Generic sub-categories for *users* and *purposes* have been identified (*buy, plan, train, install, operate, maintain*). Each user sub-category is defined in terms of what other information categories it can contain (see Figure 1). The *tasks* category enables information to be associated with the tasks that have to be performed by users. Sub-categories for *tasks* are defined by grouping similar objectives together; again, generic sub-categories have been identified (*operate, configure, describe, monitor, test, order, plan, bill, replace, group, sort, install*). The *primitives* category enables information to be associated with the type of information needed by users; generic types have been identified (*messages, alarms, services, hardware*).

The *users, purposes, tasks, and primitives* information categories are used to tag all information modules, either statically or dynamically, using, say, the Standardized General Markup Language (SGML) and information is retrieved at the appropriate point during task completion using an SGML-compliant browser. The SGML tags are used to define all hierarchical and associative relationships between each user, purpose, task, and primitive category and all information modules. All information is modular with the information categories determining module access. For existing documents, the same information source and document organization could be used for both printed and online media.

An example of what needs to be done to define the information categories, and to check the viability of the approach, is illustrated in Figures 1, 2 and 3. Figure 1 presents an information mapping of the generic information categories and sub-categories mentioned earlier. Figures 2 and 3 present two examples which show how two different users (an operator and a maintainer) would traverse the information categories to end up using the same information module (*Interpret_Alarm*).

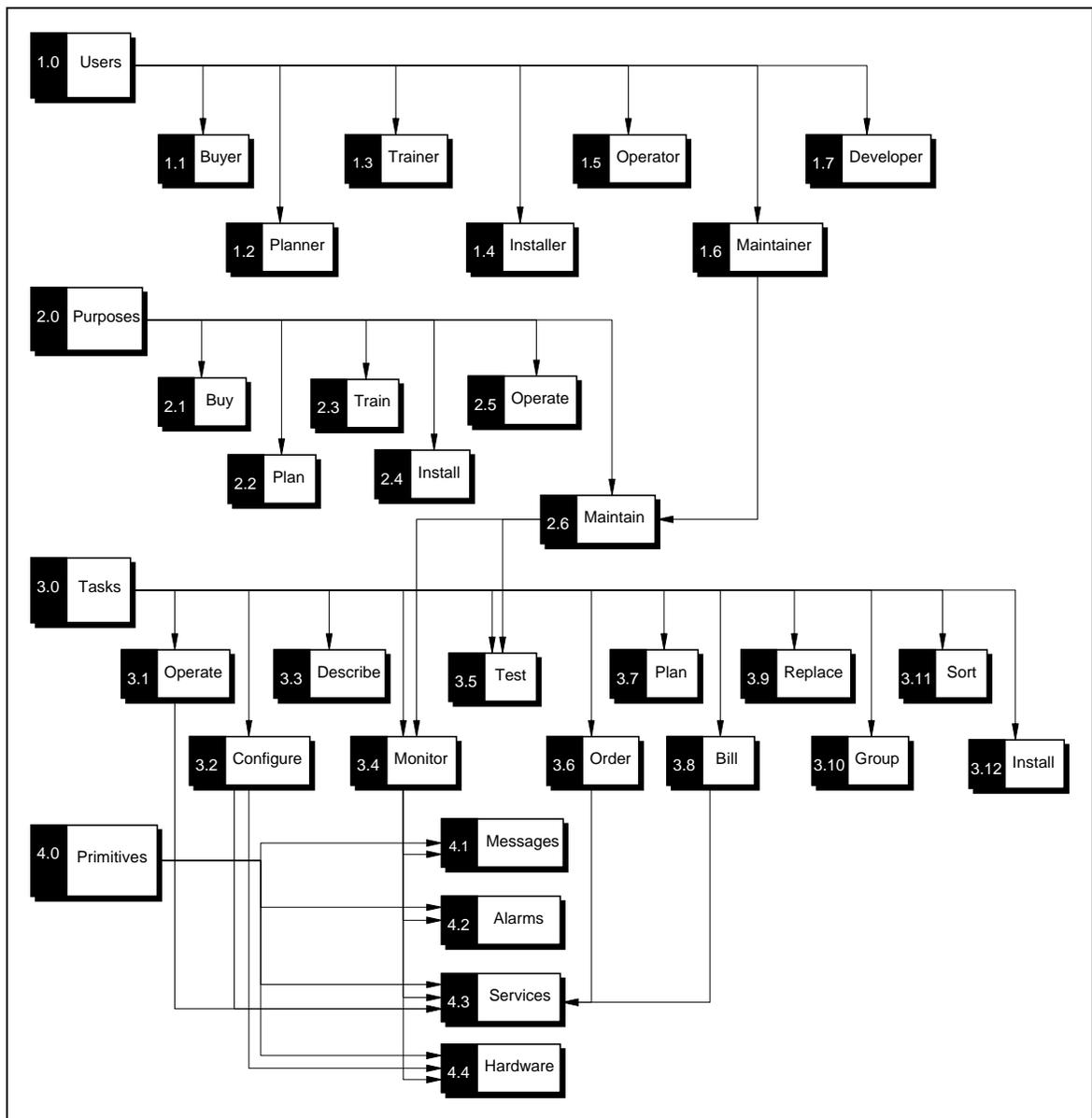


Figure 1: Generic information category model

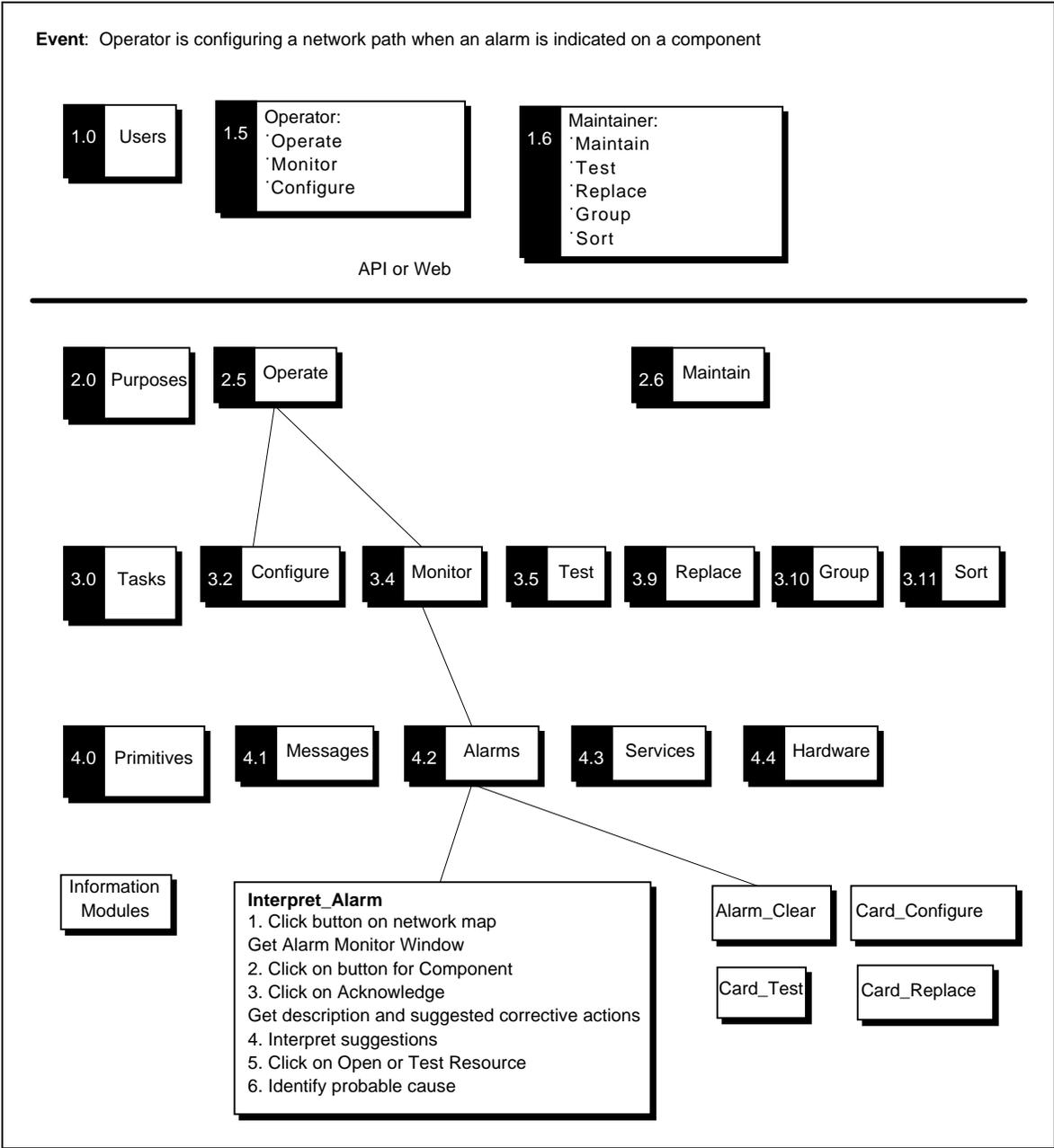


Figure 2: Interpreting an alarm example

Event: Maintainer advised that the card in shelf 2, slot 4 is dead

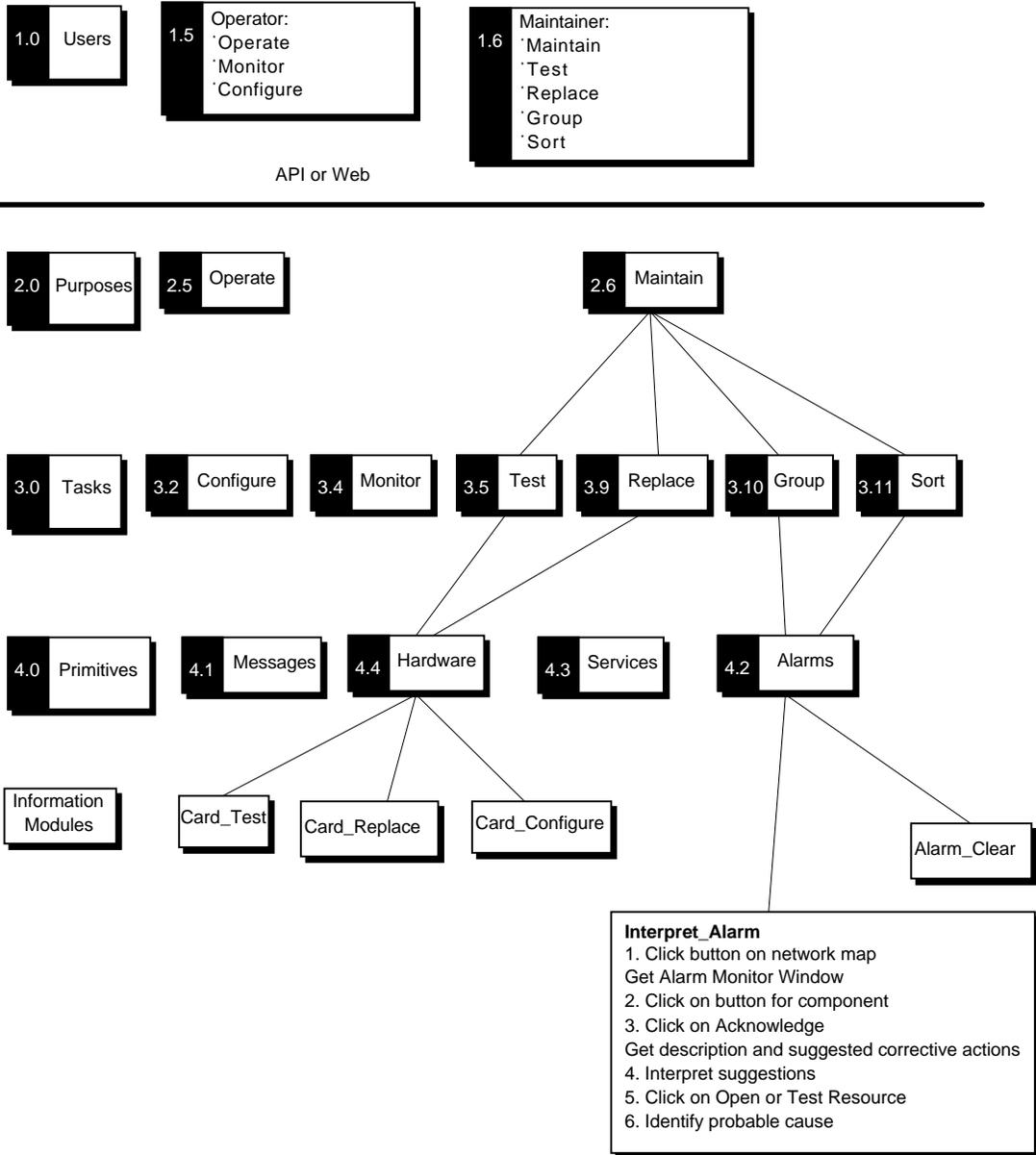


Figure 3: Replacing a card example

Conclusion

The use of a performance-based model for developing product documentation involves considering user needs from conception to completion. The resulting documentation is more likely to meet the actual, as opposed to perceived or guessed, needs of users. The model can be implemented in three stages:

- **Stage A:** Define behavioral objectives for each document chapter and organize information modules accordingly.
- **Stage B:** Categorize all objectives using the generic product information categories shown in Figure 1 and assign all information modules to one or more categories; if context sensitivity is required, use SGML to tag all information modules.
- **Stage C:** Perform usability testing using the behavioral objectives, defined in Stage A, as the evaluation criteria. Use formative usability on each information module during development and summative usability on the completed document.

The model doesn't involve anything new, it simply combines existing techniques from the documentation and training fields. The model supports multi-authorship since the effectiveness of individual information modules can be tested independently of each other. The model also fully supports online information development and the provision of specific information on a just-as-needed basis.

References

Kohn Kaminsky, Sophie. "Test Early, Test Often: A Formative Usability Kit for Writers." *ACM SIGDOC'92 Conference Proceedings* (October 1992), 47-55.

Mager, Robert F. *Measuring Instructional Intent*. Belmont: Lear Siegler, 1973.

Mager, Robert F. *Preparing Instructional Objectives*. Belmont: Pitman Learning, 1975.