Robert E. Wright

Documenting a Computer Architecture

The documentation of a computer architecture requires that special conditions be satisfied in its preparation to ensure exactness, consistency, and completeness. One group in IBM, Central Systems Architecture, has spent many years refining procedures for producing quality documentation of this type, most notably for the System/370 architecture and the System/370 Extended Architecture (370-XA). This paper describes the steps this group performs in documenting a computer architecture, the requirements identified for the finished description, and the procedures followed to satisfy those requirements.

Introduction

A computer system can have as many architectural definitions as there are recognized interfaces. The major interface is the boundary between the system's software and its hardware and microcode. This interface is defined by the computer architecture.

Within IBM, the computer architecture of each system is described in a publication called the Principles of Operation. For example, there are the *IBM System/360 Principles of Operation* and the *IBM System/370 Principles of Operation* [1]. These two publications have had a long publishing history. For System/360, publication activity for the *Principles of Operation* continued for seven years, from early in 1964 until late in 1970. Even now, orders for the 1970 edition number some 5000 copies a year. For System/370, publication activity began in the middle of 1970 and continues yet, thirteen years later.

The *IBM System/370 Principles of Operation* is important today to a great many people, both inside and outside IBM, since all machines designed as part of the System/370 family must meet its specifications, and since all machine-language programs written to operate on those machines must adhere to the interface conventions it describes. To satisfy the needs of designers and others for the detailed definition of the System/370 architecture, IBM has distributed, on the average, 100 000 copies of this publication yearly since 1970. And in 1981, when not one but two new major revisions of the publication appeared, orders exceeded 170 000 copies.

As major new architectural facilities have been introduced in System/370, the complexities associated with describing the architecture have increased, as have the number of pages needed to document it, so that what was a publication of some 220 pages at the beginning is now about 550 pages. The growth of the architectural documentation for System/370 is illustrated in Table 1.

This paper describes the way in which Central Systems Architecture, a group in IBM responsible for three major computer architectures, deals with the problems of describing a computer architecture for a sophisticated audience of computer professionals, including scientists, engineers, and programmers. First, a brief description of the steps in the documentation process of an architecture is given, followed by a review of the requirements to be met by the finished architectural description. Then, the approach the architecture group takes in meeting each of these requirements is discussed. Finally, the types of changes that might be made to an architectural description are detailed, and a description of how the architecture group deals with these types is presented.

Table 1 Growth of the *IBM System/370 Principles of Operation.*

Edition	*New facilities described*	*No. of instructions in edition*	*No. of pages†*
FIRST* (GA22-7000-0) June 1970	Time-of-day clock, control registers, extended external masking, extended I/O masking, machine-check handling, block-multiplexing control, limited channel logout, extended channel logout. (These facilities are in addition to those defined for System/360 and carried over into System/370.)	164	222
SECOND* (GA22-7000-1) Dec. 1970	Monitoring	165 (+1)	224 (+2)
THIRD (GA22-7000-2) July 1972	Dynamic address translation, extended-control (EC) mode, SSM suppression, store status and program reset, program-event recording, channel indirect data addressing, CPU timer and clock comparator	174 (+9)	278 (+54)
FOURTH (GA22-7000-3) Jan. 1973	Multiprocessing, conditional swapping, PSW key handling, clear I/O	183 (+9)	318 (+40)
FIFTH (GA22-7000-4) Sept. 1974	No functional changes; appendix of instruction-use examples added		360 (+42)
SIXTH (GA22-7000-5) Aug. 1976	Reprint of 5th edition incorporating TNL GN22-0498, which updated that edition		362 (+2)
SEVENTH (GA22-7000-6) March 1980	Recovery extensions, move inverse, invalidate page table entry, test protection, common-segment bit, low-address protection, channel-set switching	189 (+6)	426 (+64)
EIGHTH (GA22-7000-7) March 1981	Dual address space, branch and save, test block, service-signal external interruption, sense ID	204 (+15)	512 (+86)
NINTH (GA22-7000-8) Oct. 1981	Start-I/O-fast queuing, suspend and resume, 31-bit IDAWs, extended real addressing, storage-key-instruction extensions, storage-key 4K-byte block, segment protection	208 (+4)	548 (+36)

Note: In the two rightmost columns, the numbers shown in parentheses denote the number of instructions or pages added to that edition.

*The 1st and 2nd editions were issued as supplements to the *IBM System/360 Principles of Operation*, GA22-6821-8. For this comparison, the *IBM System/360 Principles of Operation* and the appropriate supplement together are considered to constitute the first and second editions of the *IBM System/370 Principles of Operation*. Thus, for those editions, the instruction count includes 151 instructions first defined for System/360, and the page count is the number of pages for GA22-6821-8 (adjusted for page-count comparisons) plus the number of pages of technical material making up the supplement.

†Because the method used in some editions to format pages results in more information being presented per page than in other editions, the actual page-number totals have been adjusted to permit true page-count comparisons from edition to edition. Specifically, the page-number totals of the 1st, 2nd, and 5th through 7th editions have been increased here by 11%.

It should be noted that the approaches described have been proven over time to work effectively for computer architecture documentation but may not necessarily be appropriate for other types of documentation.

In addition to other functions, Central Systems Architecture is responsible for the System/370 architecture, the System/370 Extended Architecture (370-XA), and the VSE architecture (the architecture that defines the ECPS:VSE mode for 4300 processors) [2]. The 370-XA architecture, which was announced by IBM in October 1981, is discussed in two other papers in this issue [3, 4].

Steps in documenting the architecture

It is important to distinguish between two major aspects of the architectural process: the definition of the architecture and the documentation of it. In creating a definition, the architecture group is primarily concerned with functional

acceptability. To this end, the group invokes a set of procedures that call for (1) the preparation of a comprehensive proposal, (2) an evaluation by all affected engineering and programming groups and by other architects, and (3) a formal ratification [5]. Although a certain level of editing is performed in order to ensure that the proposal meets general style and terminology criteria, no special effort is made at this stage to subject the proposal to a rigorous editing. The complete editing is done only after the proposal is formally ratified and incorporated. Figure 1 depicts the various steps in the creation or modification of an architectural document.

Generally, to edit before the ratification point is unproductive if the proposal is not in a form which shows it integrated with the rest of the architectural description. With few exceptions, the architecture group chooses *not* to show integration during the proposal stage because integration is time-consuming and greatly expands the amount of information that must be created and read. It delays the development of the proposal, places heavy burdens on the reviewers, and, if the proposal must be changed in any major way, is partly a wasted effort.

The real documentation phase of the architectural process, which follows ratification, focuses on making sure that the accepted definition is properly integrated into the description. The requirements for that integration are discussed in the next section.

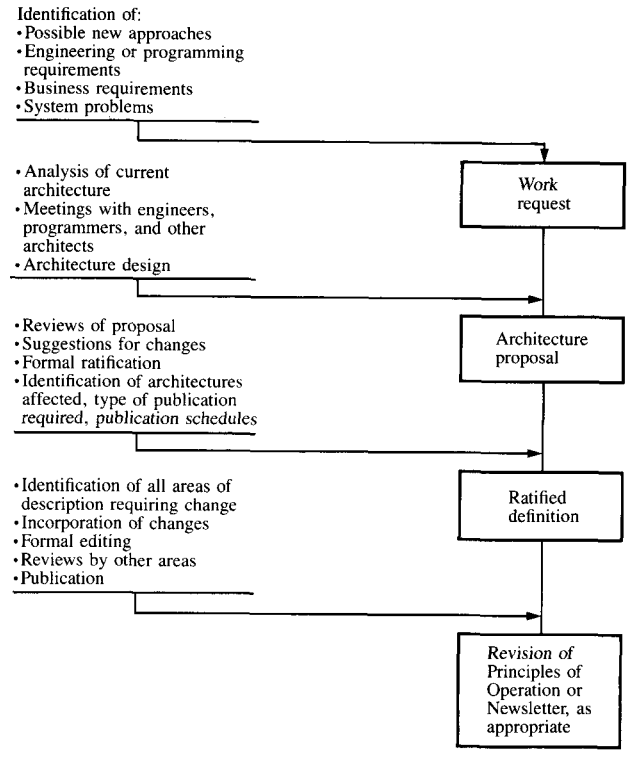

Figure 1 Steps in creating an architectural document.

Requirements to be met by the finished description

The requirements to be met by the finished description are determined in large part by the fact that the *Principles of Operation* is published, is technical writing, and deals with computer architecture. As published writing, the architectural description is distributed to a large audience. That audience ordinarily cannot consult with the author; instead, it must depend solely on its reading of the presentation for information. If the information is badly organized or presented incoherently, the message will be distorted or lost. A requirement clearly exists that the description be *understandable and well written.*

Because the description is technical writing, it is concerned with presenting practical information about technical matters. If this information is not reliable, attempts to apply it could give rise to incorrect results or even machine malfunctions. Accordingly, a requirement exists for the description to be *exact and complete.*

Finally, because the description deals with computer architecture, it should be *unified.* That is, each part should be in harmony with all other parts and with the whole, to demonstrate to the reader that the architecture itself has conceptual integrity. The existence of conceptual integrity implies that the system can be used with a high degree of ease, that is, that the design stresses economy of function and straightforwardness. Conceptual integrity as a necessary attribute of a computer architecture was identified by Dr. Frederick Brooks when he was the chief architect for System/360 and was later described by him in the book *The Mythical Man-Month* [6]. When conceptual integrity has been achieved, Glenford Myers notes in his book *Advances in Computer Architecture*, the impression is conveyed of the architecture "having been conceived, designed, and documented by a single mind" [7].

Two special requirements should also be met because the description deals with computer architecture:

- All architectural definitions should be presented in text form, not in tables or figures. This is because the definitions should be comprehensive, and tables and figures do not lend themselves to providing a comprehensive picture. (A discussion of the attributes of text, tables, and figures appears in a subsequent section, "Style guidelines.")
- Architectural definitions should be kept separate from guidelines, examples, and overviews. Guidelines, if pro-

vided, should appear as notes in the text, with appropriate headings, to distinguish them from architecture. Examples should be identified as such, perhaps by starting a sentence with the words "For example." Overviews, if they are given, should be placed in sections or chapters separate from the architecture. (An example of an overview in the *IBM System/370 Principles of Operation* is Chapter 2, "Organization"; in the *IBM 370-XA Principles of Operation*, an example is Chapter 13, "I/O Overview.") This separation should be maintained, the architecture group believes, so that both architects and users are never confused about where the architecture begins and ends.

How the requirements are met

• *Making the description understandable*

Ordinarily, if a description is generally well written, it will be understandable because good writing makes relationships clear, provides variety in sentence types, employs several methods to lead the reader from the familiar to the unfamiliar, and uses terms consistently.

But if the description uses many words that are not in the reader's vocabulary and provides few or no hints about their meaning, the description, although generally well written, may fail in making itself understood. The architecture group writes descriptions that are at the readability level of a college graduate, and comments it has received from readers generally confirm that this level is appropriate.

Three additional steps are taken to help ensure that the description remains understandable to its audience: (1) many cross-references are provided to related material in the same document, (2) special care is taken to provide a comprehensive index, and (3) constraints are placed on using terms from the general vocabulary to convey a specialized technical meaning. In this third step, every effort is made to use, instead, a compound word that conveys the appropriate technical meaning but consists of words that, individually, retain their usual meanings. For example, one term that the architecture group did take from the general vocabulary to convey a specialized technical meaning is "interruption." In its architectural sense, an interruption is a set of actions consisting in the storing of a special information unit (the current PSW) as an old PSW, the storing of information in that old PSW to identify the cause for these actions being taken, and the fetching of a new PSW to serve as the current PSW. Suppose that another set of actions similar to that characterizing an interruption needed to be defined, except that this set of actions had the additional attribute of occurring subsequent to time different from that of an interruption. One approach in labeling this other set of actions might be to pick a new term from the general vocabulary, such as "suspension," in which case that term would be lost for general use in any architectural description. Another approach, and the one favored by the architecture group, would be to look for a way to qualify the term "interruption" so that it could be used in describing both sets of actions. For example, the ordinary interruption might be described as the "primary interruption" and the other set of actions as the "secondary interruption." "Primary" and "secondary" would thus retain their general meanings except when used as part of the compound word. This approach helps lessen the probability that the reader will be confused as to whether the technical or general meaning is intended, and it helps the data processing community as a whole by preserving for general use as many words as possible.

• *Making the description well written*

Since architects vary in their writing skills, how is it possible to produce an architectural description that is well written, especially when the rules for judging excellence in technical style remain so subjective [8]? The answer has been to employ three techniques:

- Make available a style guide that treats all of the style rules to be observed and the common writing faults to be avoided, and insist that it be used.
- Use as many "canned" writing forms as possible, and encourage the "borrowing" of words, sentences, and even larger units from already published architectural descriptions. These should be descriptions that have recognized merit and that, with only minor modifications, are appropriate.
- Employ as few people as possible—preferably one architect and one editor—to handle the last few drafts of the architectural description so that one technical and one stylistic viewpoint are consistently applied.

Style guidelines

Within IBM, the general style reference to be used is the latest edition of the University of Chicago Press's *A Manual of Style* [9]. The Chicago Manual is supplemented by another style manual, only for use within IBM, that lists IBM's deviations from the Chicago Manual and treats other topics not covered by it. Even these two manuals are not sufficient, however, to deal with a number of style problems unique to architectural descriptions, and the architecture group has found it necessary to publish its own style manual, which covers most of the style information directly pertinent to architectural documentation.

In the local style manual, attention is paid of course to the more common points of style, such as punctuation, capitalization, the treatment of numbers, and cross-references. But attention is paid also to matters of tabular presentation, to the definition of terms, and to the preparation of indexes, because of the special importance of these topics in preparing architectural descriptions. For example, in the discussion of tables, emphasis is placed on the special need for clarity,

which can be achieved in part by (1) the recognition and maintenance of a grammatical relationship between the headings and entries of a table, and (2) the use of an approach that eliminates redundancy.

Few writers relish the task of index preparation. But the style manual makes the point that a good index helps the reader in four major ways: (1) it facilitates reference to a specific item, (2) it compensates for the fact that a document can be written in only one sequence and according to one plan, (3) it discloses relationships, and (4) it discloses, by the absence of certain index terms, that particular information is probably not to be found. The style manual encourages the architect, in preparing the index, to use it in ways 3 and 4 in order to identify problems in the main body of the description that might otherwise go unnoticed. Particularly because the *Principles of Operation* is written as a reference, it is not uncommon for the user reading it as a narrative to encounter terms that are not yet defined; here the index can be particularly helpful in pointing that user to another part of the description for assistance.

The use of a text-processing system will help in index preparation by automating the sorting and formatting. But index entries must still be prepared and entered manually. For help in preparing entries, the style manual refers the architect to another document, available within IBM, that concerns itself solely with the preparation of indexes. The IBM Document Composition Facility, a program product [10] which is part of the text-processing system used by the architecture group, provides for the automated part of the indexing.

One recent addition to the local style manual is a section concerned with helping architects to understand how the major elements that make up good expository writing can be used. Such writing is identified as being

- organized,
- coherent,
- at the correct audience level,
- grammatically sound, and
- consistent in style and viewpoint.

The assumption made in including this section is that, once the attributes of these elements are understood, the architect can start making informed choices. It has been the experience of the architecture group that writers with no clear picture of what choices to make often resort to following patterns they have used in the past, regardless of how inappropriate those patterns may be for the current material.

The architect is also given a discussion of the different requirements satisfied by the three key types of presentation—text, tables, and figures—so that each form will be used appropriately. Text, it is noted, allows for providing context, structure, and narrative, for making clear what relationships exist, and for identifying relative importances. None of these requirements can be met easily, if at all, in tables or figures.

If tables are used, the style manual notes, they should be used to provide summaries of definitions already stated in text. They have proved particularly useful in

- identifying multiple attributes,
- conveying similarities and differences,
- eliminating repetitiveness, and
- allowing easy referencing.

The architect is also advised that figures, such as flowcharts, have been found to be most appropriate in instances where it is important to convey, in the space of one or two pages, the relationships that exist among a number of different elements in the system. In these cases the figures do not replace a description in text but support it. They are used most effectively when a maximum compaction of information is necessary, when sequencing and branching must be conveyed, and when the material presented can be looked at selectively.

Canned forms and borrowed writing

Instruction descriptions are an example of descriptive elements that all use essentially the same form. The first paragraph in an instruction description tells generally what the instruction execution accomplishes, the next couple of paragraphs elaborate by indicating what specific mechanisms in the system are invoked, and the rest of the description details the sequence and nature of the execution. The description ends with (1) a summary of the unusual situations that can occur and (2) if appropriate, one or more programming notes. Other examples of standard forms used in the architectural description are the text format to describe the contents of fields in main storage and the table format to summarize information about instructions.

Using such forms provides for consistency in approach, allows the architect to concentrate on the subject matter and forget the writing structure, and speeds up writing and production in general.

Borrowing writing from an already published description, as long as it needs only minor modification, is encouraged. It helps three levels of individual: the writer, who then does not need to dwell on that part of the writing; the editor, who, if the published borrowing is already nicely edited, has very little work to do to make it acceptable; and the reader, who,

seeing the same words used for the same kinds of function, is reinforced in understanding the material.

• *Making the description exact*

Producing exact descriptions calls for the use of the right term, the appropriate signaling of relationships, and the use of qualifiers. To write with exactness, using these elements, it is sometimes necessary to write in structurally complex sentences [11]. In these cases, to make structurally complex sentences readable, the architecture group uses a variety of techniques:

- Parallel constructions.
- Numbering of long items of a series.
- Controlled repetition.
- Consistent hyphenation in attributive compounds.

The use of parallel constructions, in which ideas that are viewed identically by the writer are given the same grammatical structure, provides the reader with an explicit guideline for understanding how the ideas in a sentence are grouped. The numbering of long items of a series performs the same service, but in a more formal way. The use of controlled repetition, where what is said at the end of an especially complex sentence is repeated, but in a slightly different form, at the beginning of the next sentence, helps ease the journey of the reader from one sentence to the next. And the use of consistent hyphenation in attributive compounds helps make clear to the reader what grammatical groupings are intended.

Exactness is also achieved by insisting that formal English be employed. That is, no shortcuts characteristic of general and informal English are allowed: contractions are avoided, relative pronouns are not omitted, and so on. For the most part, too, the third person indicative is used. This does not mean that the writing is stiff or dull, and in fact steps are taken to guard against this possibility. But formal writing does demand concentration and presupposes in the reader an interest in the subject matter. Comments from our readers have shown that not only do they follow material that is presented in a formal style, but, on those occasions when it has been really well expressed, appreciate the style as well.

Two other elements of style used by the architecture group may in fact contribute to the suggestion of exactness, although they are not used primarily for that purpose. These elements are the use of the present tense instead of the future tense, and the use of the passive voice.

The present tense is recommended in order to simplify the presentation and to avoid the "thou-shalt" quality that exists in some specifications. It turns out that, as a secondary advantage, this approach also emphasizes that what is being described is factual.

The passive voice is also recommended in architectural descriptions because, in general, there is no reason to specify the agent of the action, which is usually the CPU or the channel. Instead, what ordinarily deserves the reader's attention in an architectural description is *what is being acted upon* [12]. A secondary advantage accrues when the careful discrimination in the use of voice is recognized to imply careful attention to all aspects of the definition.

Curiously, architects for the most part need little help in achieving exactness in their writing. The architect who is good at defining architecture is often very good at achieving a proper progression of ideas, in which qualifiers are used well, ideas subordinated appropriately, and relationships formally identified by transitional devices.

The only real problem lies in getting the architect to end a sentence at some reasonable length. The techniques just enumerated that are used by the architecture group in making complex sentences readable appear to be successful for sentences up to 80 or 90 words long, but beyond that, other actions are called for. In some descriptions that the architecture group has written, the need to make very long but exact sentences more readable has been acute, so a special approach has been employed: to replace the very long but exact sentences, a paragraph of several relatively short sentences has been created so that although no one sentence is exactly correct, the sentences of the paragraph read collectively are exactly correct. In effect, the paragraph has been made the unit that must be comprehended, not the sentence. Employing this technique does place a burden on both the architect and the editor to ensure that the result is not misleading.

Making the description complete

The architectural description must be "complete" to the extent that every function and result visible at the interface should be accounted for in the description: either the function or result should be fully described, or it should be classed as architecturally unpredictable. Unless this full accounting is striven for, the likelihood of unintentional omissions is increased. And if, because of an omission, a designer wrongly assumes that compatibility with other implementations is not required, the adverse consequences can be far-reaching.

Ensuring completeness in the architectural description is a difficult job. The responsible architect must be especially attuned to recognizing omissions. If possible, he should be helped in this checking by a technical editor with a flair for discovering inconsistencies, and also by outside reviewers with technical expertise.

One type of information that is most likely to contain omissions is lists that at one time may have been exhaustive but that, over time and after many updates, are missing

references to the newer functions. Many omissions and inconsistencies also can be found by

- Checking cross-references against their destinations.
- Checking tables against the text they support.
- Checking index terms against the cited pages.

Ordinarily, the most difficult kind of technical omission to spot is one that occurs when the description of a new function or set of functions is being integrated into the description.

If possible, whenever a major change is integrated, the architect, the editor, or both should reread the description in its entirety, adjusting all parts affected by the change as they progress through the description. The architecture group has learned never to rely on an architect's recollection of the places that must be changed; if it were to do so, some places would always be missed.

Rereading the description is considered essential in ensuring completeness. If integration activity is high and many of the changes are major, the task may actually call for the full-time services of an individual with editing talent, a good technical grasp of the subject, and a trained memory.

- *Making the description unified*

A unifying of the architectural description is achieved

- By applying quite rigid style rules.
- By using terms and phrases with extreme consistency.
- By insisting on the application of a consistent viewpoint.
- By allowing almost no redundancy.

It is also achieved by paying strict heed to two general rules:

1. In order to avoid having differences in wording that could be misinterpreted to imply differences in function,
   a. The same wording is always used to describe a particular function,
   b. The major description of a function is always included in only one place in the publication, and
   c. The architectural specification itself is always published in only one document.
2. If the formats and wording in already existing descriptions of functions work reasonably well in describing a new function, they are used. If they should work reasonably well to describe the new function but do not, they are changed so that they do.

The use of this approach ensures that new terms, new formats, and new descriptive approaches are carefully evaluated before they are included in the architectural description and are only used if the current approaches are not adequate.

Ordinarily, to achieve a truly consistent viewpoint, one architect and one editor should control the entire description. In this architecture group, however, because the CPU and channel areas each demand specialization, a single-architect approach has proved impractical. The group does try to present its descriptions so that the effect is one of a single architect having written them. To this end, one editor oversees the final wording on all major documents so that the style is made nearly the same and so that dependencies of one area on another are recognized and dealt with.

How updates are handled

The following types of change to the architectural description may at one time or another be proposed and are handled by the architecture group as indicated:

1. *Corrections of technical errors or omissions.* This type of correction is published as soon as possible. The approach previously described in the section entitled "Making the description complete" is used by the architecture group to uncover these errors and omissions.
2. *Incorporation of changes in the definition.* The change is made as soon as it is appropriate. It is ordinarily the type of change that requires a rereading of the entire description. If the change being incorporated relates to a facility that is optional on some machines, it will be necessary to provide a description of operations both with and without the facility. Also, if restrictions exist as to when the change may be published (as when the function has not yet been made available in installed machines), it is usually necessary, in updating the working copy of the description, to specify both the old and the new wording, with appropriate tags to identify when each wording applies. This approach allows for either the original wording or the changed wording to be published.
3. *Changes to meet established style requirements.* These changes are necessary, but publication is usually delayed until technical changes are required.
4. *Clarifications.* These types of changes are called for when the architect, asked to clarify a point, decides that there is merit to making a change, or when the architect recognizes the need to convey a finer shade of meaning. Publication of these changes is usually delayed until technical revisions are required. The change may require a rewording, an elaboration, or even an example.

   When the clarification is an example, it should be made clear, by the wording or by other means, that the example itself does not constitute architecture.
5. *Rewording to achieve greater consistency.* This change may or may not constitute a change to meet established style requirements (item 3). If the consistency being sought is generally conceded to enhance the reader's perception that the architecture is unified, the change is published, as specified in item 3 or at whatever time is deemed appropriate. Otherwise, the change is not made.

6. *Rewording to accommodate a particular view.* This type of change is usually proposed to correct a logical inconsistency and often involves a change in terminology. For example, when a change to the architecture causes the relationships among some elements to be changed, the use of a term may no longer be viewed as appropriate, and another term may be suggested.

   This type of change should be approached with caution because, although the logic in making the change may be sound, if the original term has been published for any length of time, all instances of it may be difficult to find and expunge, especially if other publications may have started using the term. Consequently, only the most knowledgeable architect in the group should pass on this kind of change, and usually the change should be rejected unless the perceived need to make the change is overwhelming.

   Such a change is published as soon as it is appropriate.

Conclusion

Documenting a computer architecture is complicated by the fact that not only must the ordinary requirements for good technical writing be met; in addition, the nature of computer architecture demands that the description be exact, complete, and unified. Exactness as a requirement may create additional problems in readability that are not completely solved by the usual techniques. Completeness calls for vigilance in monitoring the architectural description and can create a need for more time and manpower. Unifying the description calls for establishing many levels of consistency in the document and requires stringent control of both the style and the updating of the description. Central Systems Architecture has been successful in applying these techniques for both the System/370 and 370-XA architectures. The procedures described in this paper for controlling both style and updating to create this documentation may be useful to other architecture groups with similar objectives.

Acknowledgments

The author wishes to acknowledge the help and guidance of Andris Padegs, manager of Central Systems Architecture, whose dedication to excellence in architectural design and documentation is widely known and appreciated. Thanks are due also to Ronald M. Smith, who has been instrumental in refining many of the documentation techniques used by Central Systems Architecture. He plays a key role in the creation of CPU architecture, and his special sense of what constitutes good architecture has helped many architects and designers for many years.

References and notes

1. *IBM System/360 Principles of Operation*, Order No. GA22-6821, and *IBM System/370 Principles of Operation*, Order No. GA22-7000, available through IBM branch offices.
2. The 370-XA architecture is described in the *IBM 370-XA Principles of Operation*, Order No. SA22-7085, available through IBM branch offices, and the VSE architecture is described in the *IBM 4300 Processors Principles of Operation for ECPS:VSE Mode*, Order No. GA22-7070, also available through IBM branch offices.
3. A. Padegs, "System/370 Extended Architecture: Design Considerations," *IBM J. Res. Develop.* **27,** 198–205 (1983, this issue).
4. R. L. Cormier, R. J. Dugan, and R. R. Guyette, "System/370 Extended Architecture: The Channel Subsystem," *IBM J. Res. Develop.* **27,** 206–218 (1983, this issue).
5. The control procedure for ratification of an architectural definition is described in detail in R. P. Case and A. Padegs, "Architecture of the IBM System/370," *Commun. ACM* **21,** 73–96 (1978).
6. F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley Publishing Co., Reading, MA, 1979, pp. 42–44. Brooks contends that "conceptual integrity is *the* most important consideration in system design. It is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas." *Ibid.*, p. 42.
7. G. J. Myers, *Advances in Computer Architecture*, John Wiley & Sons, Inc., New York, 1978, p. 293.
8. Representative of the recent descriptions of studies conducted in analyzing excellence in technical-writing style is C. R. Stratton, "Analyzing Technical Style," *Technical Communication*, pp. 4–8 (1979).
9. *A Manual of Style*, 13th ed. rev., University of Chicago Press, Chicago and London, 1982.
10. Document Composition Facility Release 2, Program No. 5748-XX9, IBM Corporation.
11. Stratton asserts that, up to a point, the more structurally complex the sentence, the more effective it is for communication. C. R. Stratton, *op. cit.*, p. 5.
12. See the discussion of the "persecuted passive" in J. A. Walter, "Some Shibboleths in the Teaching of Technical Writing," *J. Tech. Writing & Commun.* **1,** 127–138 (1971).

Received August 9, 1982; revised November 29, 1982

Robert E. Wright *IBM Information Systems and Technology Group, P.O. Box 390, Poughkeepsie, New York 12602.*

Mr. Wright is the manager of architecture support for the Central Systems Architecture Department. He is responsible for establishing the style, providing editing services, and controlling the production of all major documents published by central systems architecture, particularly the principles of operations manuals for the IBM System/370, 370-XA, and ECPS:VSE architectures. Prior to joining IBM, he headed editorial groups at the Burroughs Corporation and RCA. He also held a position as editor with the Research Institute of America, a management consulting firm in New York City, where he published computer-management information for a readership of some 8000 small to medium-size companies and conducted computer-management seminars in association with the Diebold Group. He joined IBM in 1964 in Poughkeepsie as an editor with product publications, and in 1965 became a writer with that organization, working on publications for the IBM System/360 Model 91 and for the IBM System/360 architecture. In 1971, he became an editor for the systems architecture group, then part of the Systems Development Division, assuming his current management position in that group in 1978. He received a B.S. from Temple University, Philadelphia, Pennsylvania, in 1956 and pursued graduate work in communications for two years at the State University of Iowa and the University of Wisconsin. From 1961 to 1963, Mr. Wright served as an associate editor for the *STWP Review*, the national quarterly publication of the Society of Technical Writers and Publishers (presently called the Society for Technical Communication).