DEVELOPMENT OF FUNCTIONAL REQUIREMENTS FOR MULTI-FUNCTIONAL TEXT COMMUNICATION TERMINAL PHASE II

FINAL REPORT

by

D.A. MacKinnon

Prepared for the Department of Communications
Under Contract Number OST83-00206

March 29, 1984

COMPUTER GATEWAYS INC.

1, Buhr, R. J. A

2

DEVELOPMENT OF FUNCTIONAL REQUIREMENTS FOR MULTI-FUNCTIONAL TEXT COMMUNICATION TERMINAL

PHASE II

FINAL REPORT

by

D.A. MacKinnon

Prepared for the Department of Communications
Under Contract Number OST83-00206

March 29, 1984

COMPUTER GATEWAYS INC.

DOC CONTRACTOR REPORT

DEPARTMENT OF COMMUNICATIONS - OTTAWA - CANADA

COMMUNICATIONS SYSTEMS RESEARCH AND DEVELOPMENT

TITLE: DEVELOPMENT OF FUNCTIONAL REQUIREMENTS FOR MULTI-FUNCTION TEXT COMMUNICATION TERMINAL SYSTEM PHASE II

AUTHOR: D.A. MACKINNON

ISSUED BY CONTRACTOR AS REPORT NO: FINAL

CONTRACTOR: COMPUTER GATEWAYS INC.

DEPARTMENT OF SUPPLY AND SERVICES CONTRACT NO: OST83-00206

DOC SCIENTIFIC AUTHORITY: R. LEAFLOOR

CLASSIFICATION: UNCLASSIFIED

DATE: MARCH 1984

TABLE OF CONTENTS

DOCUMENT CLASSIFICATION

DOCUMENT CLASSIFICATION

1. INTRODUCTION

This report proposes a document classification scheme which may be used as a basis for determining multifunctional terminal requirements appropriate to the Canadian context. This classification may also be used as a basis for specifying conformance levels for standards developed by ISO TC97/SC18 on Text Preparation and Interchange. This classification is a synthesis and extension of the classification schemes proposed in [ISO 83] and [ISO 84b].

Section 2 presents the classification criteria and explains the rationale for their selection.

Section 3 discusses how the document classes identified in [ISO 83] (and summarized in Appendix A) fit into the proposed classification scheme and illustrates the incompleteness of this preliminary list of classes.

Appendix A contains a discussion of the classification scheme presented in [ISO 83] and identifies some of its shortcomings.

Appendix B contains a list of control characters that may be embedded in processable text. These control characters are not present in existing ISO standards such as ISO 6937 and ISO 6429, but are required to support full processability of character-box text. The list in Appendix B is a revision of Appendix F and incorporates comments and material received from international experts.

Appendix C contains a list of issues not identified in the current output documents of ISO TC97/SC18/WG3 on Office Document Architecture and Office Document Interchange Format [ISO 84a,b]. These issues require resolution before these documents can be accepted as international standards.

Appendices D,E and F contain three earlier contributions to the work of SC18 prepared in the course of this contract. These contributions have already been considered by SC18/WG3 and incorporated into the output documents of the WG3 meeting in Geneva February 1-10, 1984 [ISO 84a,b]. Note that Appendix F is an earlier version of Appendix B.

2. CLASSIFICATION CRITERIA

Document classes may be identified on the basis of three criteria:

- type of graphic architecture
- type of structure
- structural complexity

2.1 Type of Graphic Architecture

This criterion addresses the type of content found in a document. A graphic architecture consists of a repertoire (or set of repertoires) of graphic elements, a set of presentation attributes that may be associated with those elements and a set of rules for associated attributes and graphic elements, e.g.

- final form character box, as in Teletex,
- final form photographic, as in Facsimile,
- processable form character box,
- graphic display, as in the Graphical Kernel System (GKS) in ISO DS 7942,
- Videotex as in CCITT Recommendation F.300, S.100 and S.1xx (draft).

There is no limit on the number of graphic architectures which may be defined as different architectures may be specified for different applications.

2.2 Type of Structure

This criterion addresses the type of structural information present in a document. In accordance with the SC18 model for Office Document Architecture (ODA) [ISO 84a], there are four kinds of structural information which is permissible:

1) Specific Logical Structure: This structure identifies the composition of a specific document in terms of logical objects such as titles, paragraphs, footnotes, etc.

2) Specific Layout Structure: This structure identifies the composition of a specific document in terms of layout objects such as page sets, pages, frames and blocks.

3) Generic Logical Definitions: A generic logical definition acts as a template for generating specific logical objects. A generic logical definition for a document class consists of the set of generic logical definitions for the objects common to the document class.

4) Generic Layout Definitions: These act as templates for generating specific layout objects. A generic layout definition for a document class consists of the set of generic layout definitions for the objects common to the document class.

Within a document, various combinations of the above structures and definitions are permissible. Table 2.1 defines the valid combinations.

Table 2.1 Permissible Combinations of Structural Types

LOGICAL LAYOUT	NONE	SPECIFIC	GENERIC	SPECIFIC + GENERIC
NONE		X	X	X
SPECIFIC	X	X		X
GENERIC	X	X	X	X
SPECIFIC + GENERIC	X	X		X

2.3 Structural Complexity

Structural complexity refers to the degree of complexity in the structure of a document. There are many possible levels of complexity and many possible criteria from which to choose; the following four measures of complexity have been selected as useful yardsticks for determining the amount of intelligence required of an implementation to process a given document:

1) Layout Composition: The three levels of complexity associated with this measure are:
   - page only
   - page and block
   - all layout objects

2) Transparency: An implementation may or may not support the transparent overlay of text blocks.

3) Claiming: An implementation may support three levels of claiming:
   - no claiming
   - no claiming at the highest level of frame
   - claiming allowed at all levels

4) Logical Relations: Of all logical attributes, the relational ones are the most important in determining the processing complexity for a given document. The following types of relations may be supported:

- no relations
- logical-layout relations only
- logical-logical relations only
- all relations

The layout composition, transparency and claiming measures are related. Table 2.2 defines which combinations of these measures are legal.

Table 2.2 Legal Combinations of Layout-related Complexity Measures

CLAIMING / OTHER	NO CLAIMING	NO CLAIMING HIGHEST LEVEL	FULL CLAIMING
PAGE ONLY	X		
PAGE + BLOCK (OPAQUE)	X		
PAGE + BLOCK (TRANSP)	X	X	X
ALL OBJECTS (OPAQUE)	X	X	X
ALL OBJECTS (TRANSP)	X	X	X

From Table 2.2, we see that there are nine different possibilities for classifying documents, solely on the basis of layout complexity. Logical complexity is independent of layout complexity; therefore, for each layout possibility, there are four possible levels of logical complexity.

3. DOCUMENT CLASSES

3.1 Number of Classes

On the basis of the foregoing criteria, it is possible to identify as many as 291 different classes per graphic architecture , calculated as follows:

No. of document classes containing layout structure only: 3 x 9 = 27

No. of document classes containing logical structure only: 3 x 4 = 12

No. of document classes containing some combination of layout and logical structure: 7 x 9 x 4 = 252

Total Number of classes: 27 + 12 + 252 = 291

As the number of graphic architectures is unspecifiable, there is no definite limit on the total number of document classes that may be defined.

3.2 Comparison with ISO Classification

In the document classification proposed in [ISO 83], a total of 13 document classes are proposed. Within these 13 classes, there are seven different graphic architectures, namely 6937.3 level 1; 6937.3 level 2; extended 6937; CCITT

T.b; geometric (e.g. videotex or GKS/VDM); combined character, geometric and photographic; and character box with CCITT T.b.

Using the classification scheme presented above and the set of graphic architectures identified in [ISO 83], there would be a possible total of 2037 different document classes. The thirteen classes identified in [ISO 83] is clearly insufficient.

It should be noted that the classification scheme presented in this report considers each legal combination of classification criteria as a different class. It may be less overwhelming to identify a smaller set of classes based on a more restricted set of criteria, and then to use the additional criteria as the basis for identifying conformance levels within each broad class. Using this approach, the suggested initial classification criteria would be graphic architecture and type of structure. Structural complexity would then become a basis for defining conformance levels within a class. With this approach, the number of classes per graphic architecture is limited to thirteen, which is a more manageable number. For the seven graphic architectures identified in [ISO 83], there would then be only a total of 91 different classes, which is still considerably greater

than the 13 identified in [ISO 83].

APPENDIX A

THE SC18 APPROACH TO DOCUMENT CLASSIFICATION

APPENDIX A

THE SC18 APPROACH TO DOCUMENT CLASSIFICATION

1. INTRODUCTION

In their attempt to define standards for the preparation and interchange of office documents, the SC18 members came to the realization that it would be impossible to develop standards that would apply equally to all forms of documents and equipment found in an office. The proposed solution to this difficulty was to identify various broad classes of documents, with varying levels of complexity within each class. As standards are developed, conformance clauses will indicate to which classes and levels of document each part of a standard applies.

In this way, developers of protocols and equipment are not constrained to implement sophisticated and expensive features which are not needed for a particular application; they need only implement the capabilities associated with the document class(es) closest to the chosen application.

The problem faced by SC18 is identical to the one faced by Canadian designers of multi-functional terminals. Since the work of SC18 includes both character-box text, facsimile and geometric text (including videotex), the document

classification scheme that will be eventually be adopted by SC18 should be directly applicable to the Canadian environment.

In the following, a brief review is provided of the document classification scheme currently proposed by SC18. This is followed by a discussion of the areas where further work is required before a sufficiently detailed set of document classes can be identified.

2. SC18 DOCUMENT CLASSIFICATION SCHEME

The classification scheme currently proposed by SC18 is based on the following criteria:

1) nature of the content of the text, e.g. whether it is character-box, facsimile, geometric or a mixture of the above.

2) complexity of the content, i.e. the architecture used to represent the content, e.g. the character set for character-box text.

3) nature of the intended use, e.g. whether the document is to be imaged only, or can be processed after interchange.

4) complexity of the intended use, e.g. whether the permitted processing includes changing the logical or layout structure of the document.

On the basis of the above criteria, the following thirteen classes of documents were identified:

CLASS NAME	CHARACTERISTICS
Character Box Page Image Format PIF-1	Contains only character-box text with format effectors belonging to ISO 6937.3 Level 1; contains no logical structure and is intended only for imaging.
Character Box Page Image Format PIF-2	Contains only character-box text with format effectors belonging to ISO 6937.3 Level 2; contains no logical structure and is intended only for imaging.
Character Box Processable Format TPF-1	Contains only character-box text containing extended format effectors (a superset of the PIF-1 character set); contains no logical structure and is intended for limited processing of the layout format.
Character Box Processable Format TPF-2	Contains only character-box text containing no format effectors; contains logical structure but no layout structure; intended for processing of the logical structure and formatting according to the wishes of the recipient.
Character Box Processable Format TPF-3	Contains only character-box text containing extended format effectors (a superset of the PIF-2 character set); contains both logical and layout structure and is intended for

	full processing after interchange.
Photographic Image Format	Contains only photographic text encoded using CCITT T.b (Group 4 facsimile); contains no logical structure and is intended only for imaging.
Graphic Format	Contains geometric text using a "unit screen" architecture, e.g. videotex or GKS/VDM. No logical structure; intended only for imaging.
Combined Format	Contains a mixture of character-box, photographic and geometric text, using a "unit screen architecture". No logical structure.
Mixed Image Format MIF-Ø	Contains both character-box and photographic text; contains similar format effectors as PIF-1; no logical structure and a layout structure consisting of single-mode pages; intended for imaging only.
Mixed Image Format MIF-1	Contains both character-box and photographic text; contains similar format effectors as PIF-2; no logical structure and a layout structure which includes both mixed-mode pages and blocks; intended for imaging only (equivalent to TIF-1 in CCITT S.a).
Mixed Processable Format MPF-1	Contains both character-box and photographic text; contains extended format effectors as in TPF-1; has full layout structure but no logical structure; intended for processing of the layout structure after interchange.
Mixed Processable Format MPF-2	Contains both character-box and photographic text; contains logical structure as in TPF-2 and possibly a limited layout structure (frames) if layout directives are present;

	intended for processing of the logical structure after interchange.
Mixed Processable Format MPF-3	Contains both character-box and photographic text; contains both full logical and layout structures and is intended for full processing after interchange.

The above classification is incomplete in the following ways:

1) it does not include geometric text within any mixed-mode class. This omission is a reflection of the immediate concerns of SC18 which are directed principally toward the compatibility with the work of CCITT (Recommendation S.a); this work focuses on interchange of mixed-mode character-box and photographic text where the layout structure is processable. To this date, not enough study has been given to the interchange of text with processable logical structure.

2) it does not identify which extended format effectors are to be included in classes TPF-1, TPF-2, TPF-3, MPF-1, MPF-2 and MPF-3. It is likely that multiple sets of format effectors will be defined, leading to multiple levels of conformance within these classes.

3) additional classes may be required for page-image format text to include format effectors not identifed in ISO 6937.

In addition to the above limitations, the SC18 classification scheme seems inconsistent in its definition of classes TPF-2 and MPF-2. The former includes no layout structure while the latter makes a provision for the inclusion of frames when the logical structure includes layout directives. As frames are an optional component of the layout structure, it seems inappropriate to associate layout directives only with frames. For consistency with TPF-2, MPF-2 should not include any layout structure.

3. CONCLUSIONS

The SC18 document classification scheme provides a sound foundation for the development of a detailed document classification scheme. However, it must be recognized as a preliminary classification; further research is required in the following areas to improve its completeness and to maximize its usefulness:

1) identify and categorize format effectors which control the processing of text.

2) identify additional mixed-mode document classes which include geometric text.

3) within the individual classes, identify levels of complexity which could serve as the basis of conformance clauses in the evolving SC18 standards and which could assist in more precisely defining functional requirements for multi-functional terminals.

APPENDIX B

ADDITIONAL FORMAT EFFECTORS

FOR PROCESSABLE TEXT INTERCHANGE

1. INTRODUCTION

The "unofficial" meeting report of the ad-hoc group on functional capabilities of documents, in Recommendation number 5, urged that priority work for both WG3 and WG5 should be the development of logical and layout structures for processable text forms. This work is to yield TPF and MPF levels 1,2 and 3, as described in Annex A of that document.

In support of this objective, this contribution addresses the need to define additional format effectors beyond those defined in existing standards (ISO 6937, ISO 6429, ANSI X3.64, ANSI X3.98). It is generally recognized that the set of format effectors defined in these standards is incomplete and inadequate for the interchange of processable text.

Within the ODA, it is possible to define logical and layout structures which subdivide the text into arbitrarily small content portions. For example, it is legitimate to define a layout structure which associates a different layout block with each word of the text; similarly, the corresponding logical structure could also define the word as the smallest unit of structure. This approach allows

almost all information relating to how the text is to be processed to be defined in the form of attributes of these structural components. There would be little or no need for embedded control characters within content portions. While this approach is possible, it is unwieldy and would lead to high transmission overhead during interchange. It is therefore important to be able to embed format effectors within content portions to provide control over the processing of content portions that are smaller than those defined in the logical or layout structure.

The format effectors defined here are limited to those which control the formatting and imaging of character-box text;

2. FORMAT EFFECTORS

The format effectors are organized into three categories:

Category 1: These format effectors are common to most processable documents. They apply to both soft and hard copy displays.

Category 2: These format effectors are less common and would be used in more sophisticated text processing environments. They apply to both soft and hard copy displays.

Category 3: These format effectors apply only to imaging on hard copy devices.

The format effectors presented here include some proposed in the report of the ad-hoc group on functional capabilities of documents, in WG3 N133 ("Draft Proposed American National Standard For Basic Processable Text Interchange Format"), in WG3 N26Ø ("Office Document Architecture for Editable Text" by Ted Richards of AES Data Inc.), in the NBS Report "Document Interchange Format" by Joan Knoerdel and Shirley Ward Watkins, plus others identified in the course of a study of word processors available in the Canadian market.

Format effectors defined in existing standards are not included.

Delimiters which identify the beginning and end of text blocks (e.g. the begin/end text mark characters proposed in WG3 N133 for delimiting margin text) are not included here for these are more properly attributes of individual content portions. Similarly, margin settings are considered to be attributes of structural objects and are not considered here.

2.1 Category 1 Format Effectors

inserted end of line: indicates an end of line that has been inserted in the middle of a word during a formatting operation.

space end of line: a space that has been turned into an end of line during a formatting operation. It is treated as an end of line by the imaging software and as a space by editing software. It will revert to a normal space if the word ceases to be at the end of a line.

inserted end of page: indicates a page boundary inserted as a result of a pagination operation, but which may be removed during subsequent repagination.

inserted hyphen: indicates a hyphen inserted during formatting when a word is hyphenated at the end of a line. It is imaged as a normal hyphen. (Also called a "soft" or "syllable" hyphen).

inserted space: a space character inserted by the formatting software during justification. It will be discarded immediately before the next formatting operation.

connecting space: indicates a space which cannot be extended during justification and which cannot be used as a line boundary. (Also called a "non-line-breaking" or "hard" space).

invisible space: a non-printing space character. When a line break is created between two words separated by more than one space, the formatting software changes one of the spaces into a "space end of line" character and the rest into invisible spaces in order to allow proper alignment at the beginning of the next line. They revert to normal spaces if the first word ceases to be at the end of a line. The invisible spaces may either precede or follow the line break.

set page number value: indicates the value that the next page number will take when printed. (Note that this is more general than setting a page numbering initial value).

set page numbering: turns page numbering on or off.

2.2 Category 2 Format Effectors

ghost hyphen: indicates a point in a work where the word may be hyphenated, if necessary. This is normally not displayed, but is displayed if the hyphenation does occur. (Also called a "syllable hyphen point" or "break permitted here").

non-line-breaking character: indicates that the subsequent character must not appear at the end of a line during reformatting (also called "no break here").

begin/end nospace underline: delimits text to be underlined, excepting spaces.

begin/end nospace double underline: delimits text to be double underscored, excepting spaces.

begin/end note: delimits text to be treated as comment and suppressed during output.

tab block stop: indicates a location to which the active position will be moved during a block tab.

tab block insert: a tab block stop which will leave the system in insert mode after a block tab.

merge variable delimiter: delimits successive merge variables in a merge group to be used during a merge.

merge with insert: identifies the location at which a merge variable will be inserted during a merge.

merge with overwrite: identifies the location at which a merge variable will be written, possibly overwriting old text, during a merge.

begin/end ignore during merge: delimits text to be omitted during a merge such as an operator's note.

ignore merge variable: causes the next merge variable in a variable list to be skipped during a merge.

macro: causes the specified macro expansion to be inserted into the text at the indicated position.

print current date: causes the system date to be printed at the indicated position.

NOP: causes subsequent control function to be disabled (i.e. no action taken) during imaging.

NOE: causes subsequent control function to be disabled during editing but not during imaging.

replace by space: causes subsequent format effector to occupy a position on output and be printed as a space.

column backspace: causes the active position to move to the preceding column on a row, unless it is at the first column in which case there is no effect.

zero motion: causes the normal horizontal motion after imaging the subsequent character to be suppressed, causing a second subsequent character to be imaged at the same position as the first.

set tab replacement character: determines the character to be imaged during a tab function.

extend line: releases the right margin, allowing text to be imaged beyond the end of the line.

return to current line: causes active position to move to the line at which it was prior to vertical movement.

indent n MUs: causes text to be indented by the number of measurement units (MU's) specified. The exact value of a measurement unit is assumed to be defined elsewhere in the interchanged document.

stop indent: terminates indentation.

set strikeout character: determines strike-out character to be used.

alignment tab: causes the active position to move to the column specifies as an alingment tab stop and text to be entered to the left of the stop.

unknown character: contains a character which is not part of the standard character set. It must allow for a multi-byte sequence to encode characters from a 16-bit (or larger) character set. The character will be preserved by processing programs, but must be imaged as a single character, perhaps "?".

variable length invisible space: a variable length space which is temporarily 'between' lines (see invisible space above).

2.3 Category 3 Format Effectors

begin/end proportional printing: delimits characters which will each occupy a variable amount of space when printed.

suspend printing: causes printing to be suspended until resumed by operator.

line overstrike: causes the subsequent line to overwrite the previous line (i.e. disables the normal vertical motion of the printer prior to printing the subsequent line).

numeric space: causes a space to be printed having the same width as a single digit in the current pitch.

unit backspace: moves the printing point left one escapement unit in proportionally spaced type styles. Treated as a backspace overwrite in monospaced type styles.

partial unit backspace: moves the printing point left zero to one escapement unit in proportionally spaced type styles.

numeric backspace: moves the printing point left a distance equivalent to the escapement value for digits in the current pitch.

BIBLIOGRAPHY

1. ANSI X3.64, "Additional Controls for Use with American National Standard Code for Information Interchange", July 1979.

2. ANSI X3.98, "Text Information Interchange in Page Image Format (PIF)", 1983.

3. ISO IS 6429, "Additional Control Functions for Character-Imaging Devices", May 1983.

4. ISO DIS 6937, "Information Processing - Coded Character Sets for Text Communication - Part 3: Control Functions for Page-image Format", 1983.

5. ISO TC 97/SC 18/WG 3/N 133, "Draft Proposed American National Standard for Basic Processable Text Interchange Format", March 1983.

6. ISO TC 97/SC 18 Unofficial Report of Meeting of Ad-hoc Group on Functional Capabilities of Documents, November 1983.

7. ISO TC 97/SC 18/WG 3/N 260, "Office Document Architecture for Editable Text", Ted Richards, AES Data Inc.

8. Knoerdel, Joan E., and Shirley Ward Watkins, "Document Interchange Format".

9. Systemhouse Limited, "A Standard Protocol for Allowing Text Preparation Terminals to Create, Store and Interchange Text in a Compatible Manner", Department of Communications - Canada, November 1983.

APPENDIX C

UNRESOLVED TECHNICAL ISSUES NOT IDENTIFIED IN CURRENT ISO TC97/SC18/WG3 ODA AND ODIF WG3/N283 AND WG3/N284

TECHNICAL ISSUES NOT IDENTIFIED
IN ISO TC97/SC18/WG3 ODA AND ODIF
ISO TC97/SC18/WG3 N283 AND 284

A. Document Processing Model

1. What is the precise relationship between the document profile and the document body? If every change to a document creates a new logical/layout structure, does this result in a new document? If so, must the document profile be updated (with a new version number) as well?

2. What is the precise role of frames and blocks in the layout structure? Are they both fixed size, or can one or both of them be variable?

3. What is the role of the specific layout structure? Can it be edited directly, or is it inaccessible to the user? In one view, it is the output of the formatting process and can only be changed by changing the specific logical structure or the logical or layout definitions. In another view, direct manipulation of the layout structure is possible (e.g. moving an arbitrary chunk of text around on the screen directly alters the specific layout structure); in this case, some unspecified

mechanism is needed to update the logical structure so that all structures remain consistent.

4. When must the logical and layout structures be consistent? At all times or only during interchange?

B. Office Document Architecture (ODA)

1. Is the sequence constructor redundant since the order of subordinate objects defined in the logical structure has the desired effect?

2. Can object definitions contain attributes which pertain only to themselves and not to the specific objects which they generate? If so, can these attributes conflict with attributes of corresponding specific objects?

3. Do attribute values permit rules for their calculation or are they restricted to fixed values?

4. Is it possible within the logical or layout structure to have different object types at the same level in the hierarchy? For example, is it sensible to permit a page to have as its immediately subordinate objects a block and a frame?

5. Is the current concept of conceptual page in combination with the "medium type" attribute sufficient to permit multiple pages to be imaged on one medium? For example, it would be desirable to take a 10-page report which normally is printed on 10 paper pages and compress the entire report onto a single microfiche. In this case, a single physical page contains 10 conceptual pages.

6. Is there a need for layout-logical relations to handle relationships between layout and logical objects? A possible example is the mapping of page numbers into tables of contents and indexes.

7. The precise effect of claiming frames on underlying blocks is still not clear.

8. There is a need to identify additional graphic architectures to be supported by the ODA standard. Currently, only two architectures are defined - one a character-box architecture based on ISO 6937 and the other a photographic architecture based on CCITT Recommendation T.b. In particular, graphic architectures supporting geometric text and processable character-box text need to be identified.

9. The precise role of generic blocks requires clarification. Currently, they are permitted only if they contain common content. Is this restriction appropriate or is there some use in defining generic blocks without content?

10. Do the margin and tabulation attributes apply only to frames, only to blocks or to both? What is their precise role? Should it be possible to associate these attributes with logical objects?

11. Should it be possible to define multiple sets of margins within a frame? This would permit multiple columns of text to be placed within a single frame; currently, multiple columns are to be handled by defining separate frames for each column.

12. Currently, layout directives map a single logical object into a layout object. Is there a need to identify layout directives which define how related logical objects should be laid out? A possible example is a directive indicating that a figure is to be placed after the first reference to it.

13. Currently, logical-to-logical relations such as reference to footnote or reference to figure are handled as special cases of a powerful, general-purpose "calculation rule" attribute; this was done because no use for such references was identified other than for ensuring consistency of numbering schemes. Are there any valid reasons for separating such references as a separate type of attribute?

C. Office Document Interchange Format (ODIF)

1. The current ODIF document permits the text units to be transmitted in any order. Is this flexibility useful or would it be more appropriate to impose an order on the text units?

2. Are the preliminary criteria for defining conformance levels for ODA/ODIF documents appropriate? Should the number of conformance levels be large or small?

3. Does SGML have a useful role to play as an interchange format for office documents? Is it possible to merge specific logical and layout structures in an SGML data stream? Is it useful to consider the use of SGML for representing only the logical structure and the current ODIF for the layout structure, resulting in a hybrid

data stream?

D. Other

1. Is the current organization of the WG3 documents appropriate? Should the ODA model be separated from the attribute definitions and be treated as a separate document, along the lines of the SC16 Reference Model for Open Systems Interconnection? Or should an overview of the ODA model be provided in the General Introduction of the standard (Part 1), with the current ODA document remaining substantially as is?

APPENDIX D

COMMENTS ON WG3 N 228, OFFICE DOCUMENT ARCHITECTURE

(PROPOSED FIFTH WORKING DRAFT)

ISO/TC 97/SC 18/WG3 N 271

TITLE: Comments on WG3 N 228, Office Document Architecture (Proposed Fifth Working Draft)

CONTRIBUTOR: D. MacKinnon, R. Leafloor

STATUS OF DOCUMENT: Canadian Expert Contribution

PURPOSE OF CONTRIBUTION: To propose improvements to the above document. These comments are in general applicable to N207 as well.

1) p. 6: the term "presentation objects" should be replaced by "layout objects", for consistency with the remainder of the document.

2) p. 7: the text "(eg standards specific to character, geometric or photographic elements)" should be replaced by "(eg standards specifying character, geometric and/or photographic elements)".

As it is, it implies that these standards are specific to a single category of elements where several of them provide for more than one type.

3) p. 8: replace the following text on the common properties of document classes,

" - the types of logical text objects that may occur and relationships between them;

- predefined layout objects;

- common portions of content;"

with the following:

- the types of logical and layout text objects that may occur and relationships between them;

- the graphic sub-architecture for portions of the document content;

This change brings the basis for document classification in line with that used by the Geneva Ad-Hoc Group on Functional Capabilities of Documents. Note that predefined layout objects and common portions of content are not included in the classification criteria for document classes. They are more appropriately used as the basis for defining generic documents within a particular class.

4) p. 8: in the NOTE: at the bottom of the page, add "or content" after the word objects.

5) p. 9: The content of a basic logical text object should be of a single graphic sub-architecture and not of a single category of graphic element. In this way, a consistent view of the content portions of a document can be maintained, regardless of which structure is being used to

view the document.

6) p. 11: 2nd paragraph: Extend the last sentence to include

> "The structure allows for predefined hierarchical and organisational relationships between logical text objects, possibly with predefined contents or attributes for some of the basic text objects, e.g. standard paragraphs with standard layout directives."

7) p. 12: the first paragraph of 5.3.1 states that terminal nodes consist only of basic layout objects; does this preclude a branch ending in a null frame? How does one reserve blank areas in the layout structure?

8) pp. 14 and 15: replace the expression "single category of graphic element" with "single graphic sub-architecture".

9) p. 15: A generic block need not have predefined content - it could simply define an area within the image area where text is to be placed - and if it does contain predefined content, this content

need not be associated with a document class; it could simply be associated with a group of documents which share the common content.

10) p. 15: according to the definitions on page 14, the simplest generic structure consists of a generic page. The generic frame could not be part of the simplest structure because it is always an optional component of the structure.

11) p. 20: Section 6.1 b) item 4); add "basic" before objects, for these are the only objects which have content.

12) p. 32: The type of sub-architecture should be a separate attribute rather than part of "content type".

13) p. 32: The paragraph beginning with "In the initial version of this standard ..." should be deleted. Geometric graphic elements need not be excluded at this stage, as they have been included in the work of WG5 and are present in the document classes proposed by the Geneva Ad-Hoc group.

APPENDIX E

LAYOUT AND PRESENTATION DIRECTIVES

ISO/TC 97/SC 18/WG3 N 270

TITLE: Layout and Presentation Directives

CONTRIBUTOR: D.A. MacKinnon

STATUS OF DOCUMENT: Canadian Expert Contribution

PURPOSE OF CONTRIBUTION: A revised and extended version of sections 6.7.2 and 6.7.3 of WG3 N228. The principal revision is the removal of the restriction that layout directives apply only to frames. The extensions consist of additional layout directives and new presentation directives.

Frames are optional components of the layout structure; hence, it is illogical to allow layout directives to refer only to frames. Blocks and pages may also be referred to. Further, layout directives need not refer explicitly to any layout structure; it is possible to interchange a document having only logical structure and layout directives which do not refer to any specific layout object.

In the following, revised versions of sections 6.7.2 and 6.7.3 of N 228 are proposed. These include directives not previously mentioned.

6.7.2 Layout Directives

A layout directive of a composite or basic logical text object guides the generation of the shape and the placement of the text image of the contents of that object. The placement of the text image may be specified in relation to an explicit page, frame or block (when the layout structure is pre-defined) or in relation to the current layout object (when no layout structure is pre-defined). The layout directives are taken into account by the formatting operation that creates the blocks to represent the text image. The formatting operation allocates the

contents of the logical text object to these blocks and determines the layout attributes (including the dimensions and the positions) of these blocks in accordance with the specified layout directives of the logical text object.

The following layout directives may include a

Layout Object Reference

which identifies the page, frame or block which is the subject of the associated layout directive.

6.7.2.1 Replication

(ALL) Optional: Default: No Constraint

This attribute specifies that corresponding to each instance of this logical text object is an instance of a replicatable layout object.

6.7.2.2 Centre

(ALL) Optional: Default: No Constraint

This attribute specifies that the text image is to be centered within the referenced layout object. In the absence of an explicit reference, the text image is centered within the current page.

This centering can be either horizontal, vertical or both.

(Other forms of centering are for further study).

6.7.2.3 Indivisible

(ALL) Optional: Default: No Constraint

This attribute specifies that the inferior objects are all to appear within the single specified layout object. In the absence of an explicit reference, the inferior objects are all to appear within the current page.

6.7.2.4 Balance

(FRAME or BLOCK) Optional: Default: No Constraint

This attribute specifies that the logical text object is to be placed within the specified set of flow linked frames and the amount of content to be placed within each frame should be approximately equal.

This attribute can be used only in conjunction with a predefined layout structure.

6.7.2.5 Alignment

(ALL) Optional: Default: No Constraint

This attribute specifies another logical text object to which this object is to be aligned horizontally or vertically after layout generation.

6.7.2.6 Same Layout Object

(ALL) Optional: Default: No Constraint

This attribute specifies another logical object which is to be presented in the same layout object as this object after layout generation, e.g. footnote.

6.7.2.7 Indentation

(ALL) Optional: Default: Zero Indentation

This attribute specifies an amount of indentation on the left or right of one or more text lines relative to the edges of the reference layout object. The indentation is specified either in relation to tabulation stops or in units of character box size.

In the absence of a specific reference, the indentation is performed in relation to the current layout object (typically the page).

Examples of the use of indentation are to indicate the amount that the first line of a new paragraph is to be indented, or to offset a quotation.

(It is for further study whether the logical structure should be kept independent of physical measurement units).

6.7.2.8 Border Notation

(ALL) Optional: Default: No visible border

This attribute specifies the way in which the borders of the reference layout object are to be marked.

6.6.2.9 Separation

(BASIC LOGICAL TEXT OBJECT) Optional:

Default: No Separation

This attribute specifies tha amount of vertical separation to leave between adjacent logical objects at the same level. It may be specified in terms of number of text lines or in basic measurement units.

6.7.2.1Ø Widow Size

(BASIC LOGICAL TEXT OBJECT) Optional:

Default: 2 lines

This attribute specifies the minimum number of lines of information to allocate to each layout object when the logical object must be split between two layout objects during formatting, e.g. at a page boundary.

6.7.2.11 New Page

(ALL) Optional: Default: No Constraint

This attribute specifies that the associated logical object shall be placed on a new page during formatting.

6.7.3 Presentation Directives

The presentation directives are layout directives that do not refer to layout objects.

6.7.3.1 Justify

(ALL) Optional: Default: Left Justification

This attribute specifies that the text image is to be justified within the current layout object. Possible values are left and right justification.

6.7.3.2 Wordwrap

(ALL) Optional: Default: Wordwrap On

This attribute specifies whether wordwrap is to be performed at the boundary of the current layout object. The possible values are on or off.

6.7.3.3 Hyphenate

(ALL) Optional: Default: Off

This attribute specifies what hyphenation algorithm shall be used in conjunction with wordwrap. The possible values are "off", "at hyphens only", "before certain suffixes" and "between syllables".

6.7.3.4 Set Enumeration Scheme

(ALL) Optional: Default: No Numbering

This attribute determines the numbering scheme (either arabic, alphabetic uppercase, alphabetic lowercase or roman) to be used for numbering logical objects at this level.

6.7.3.5 Set Base Number

(ALL) Optional: Default: One

This attribute specifies the base number to be used in numbering logical objects at the specified level.

6.7.3.6 Set Number Increment

(ALL) Optional: Default: One

This attribute specifies the value by which the logical object number for the level indicated is incremented.

6.7.3.7 Enumeration

(ALL) Optional: Default: No Enumeration

This attribute invokes or suppresses the printing of logical block numbers for the indicated level.

APPENDIX F

FORMAT EFFECTORS AND LOGICAL OPERATORS

FOR PROCESSABLE TEXT INTERCHANGE

ISO/TC 97/SC 18/WG 3 N 280

TITLE: Format Effectors and Logical Operators for Processable Text Interchange.

CONTRIBUTOR: D.A. MacKinnon

STATUS OF DOCUMENT: Canadian Expert Contribution

PURPOSE OF CONTRIBUTION: To address the need for additional format effectors beyond those defined in ISO 6937. It is generally recognized that the set of format effectors defined in ISO 6937 is incomplete and inadequate for the interchange of processable text. In addition to format effectors, this contribution identifies logical operators which define relationships between logical objects which may be at different logical levels.

1. INTRODUCTION

The "unofficial" meeting report of the ad-hoc group on functional capabilities of documents, in Recommendation number 5, urged that priority work for both WG3 and WG5 should be the development of logical and layout structures for processable text forms. This work is to yield TPF and MPF levels 1,2 and 3, as described in Annex A of that document.

This contribution addresses the need to define additional format effectors beyond those defined in ISO 6937. It is generally recognized that the set of format effectors defined in ISO 6937 is incomplete and inadequate for the interchange of processable text.

Within the ODA, it is possible to define logical and layout structures which subdivide the text into arbitrarily small content portions. For example, it is legitimate to define a layout structure which associates a different layout block with each word of the text; similarly, the corresponding logical structure could also define the word as the smallest unit of structure. This approach allows all information relating to how the text is to be processed to be defined in the form of attributes of these structural

components. There would be no need for embedded control characters within content portions; the content would consist solely of pure text. While this approach is possible, it is unwieldy and would lead to high transmission overhead during interchange. It is therefore important to be able to embed control characters within content portions to provide control over the processing of content portions that are smaller than those defined in the logical or layout structure.

Two types of control characters are defined here:

format effectors which control the procesing of character-box text;

logical operators which define relationships between logical objects containing character-box text.

2. FORMAT EFFECTORS

The format effectors are organized into four categories:

Category 1: These format effectors are common to most processable documents. They apply to both soft and hard copy displays.

Category 2: These format effectors are less common and would be used in more sophisticated text processing environments. They apply to both soft and hard copy displays.

Category 3: These format effectors apply only to imaging on hard copy devices.

Category 4: These format effectors apply only to imaging on soft copy devices.

The format effectors proposed here include some proposed in the report of the ad-hoc group on functional capabilities of documents, in WG3 N133 ("Draft Proposed American National Standard For Basic Processable Text Interchange Format"), in WG3 N??? ("Office Document Architecture for Editable Text" by Ted Richards of AES Data Inc.), plus others identified in the course of a study of word processors available in the Canadian market. Delimiters which identify the beginning and end of text blocks (e.g. the begin/end text mark characters proposed in WG3 N133 for delimiting margin text) are not included here for these are more properly attributes of individual content portions.

2.1 Category 1 Format Effectors

inserted end of line: indicates an end of line that has been inserted in the middle of a word during a formatting operation.

space end of line: a space that has been turned into an end of line during a formatting operation. It is treated as an end of line by the imaging software and as a space by editing software. It will revert to a normal space if the word ceases to be at the end of a line.

inserted end of page: indicates a page boundary inserted as a result of a pagination operation, but which may be removed during subsequent repagination.

inserted hyphen: indicates a hyphen inserted during formatting when a word is hyphenated at the end of a line. It is imaged as a normal hyphen. (Also called a soft or syllable hyphen).

inserted space: a space character inserted by the formatting software during justification. It will be discarded immediately before the next formatting operation.

connecting space: indicates a space which cannot be extended during justification and which cannot be used as a line boundary. (Also called a non-line-breaking space).

invisible space: a non-printing space character. When a line break is created between two words separated by more than one space, the formatting software changes on of the spaces into a "space end of line" character and the rest into invisible spaces in order to allow proper alignment at the beginning of the next line. They revert to normal spaces if the first word ceases to be at the end of a line. The invisible spaces may either precede or follow the line break.

2.2 Category 2 Format Effectors

ghost hyphen: indicates a point in a work where the word may be hyphenated, if necessary. This is normally not displayed, but is displayed if the hyphenation does occur. (Also called a syllable hyphen point).

non-line-breaking character: indicates that the subsequent character must not appear at the end of a line during reformatting.

begin/end nospace underline: delimits text to be underlined, excepting spaces.

begin/end nospace double underline: delimits text to be double underscored, excepting spaces.

begin/end note: delimits text to be treated as comment and suppressed during output.

tab block stop: indicates a location to which the active position will be moved during a block tab.

tab block insert: a tab blick stop which will leave the system in insert mode after a block tab.

merge variable delimiter: delimits successive merge varaibles in a merge group to be used during a merge.

merge with insert: identifies the location at which a merge variable will be inserted during a merge.

merge with overwrite: identifies the location at which a merge variable will be written, possibly overwriting old text, during a merge.

begin/end ignore during merge: delimits text to be omitted during a merge such as an operator's note.

ignore merge variable: causes the next merge variable in a variable list to be skipped during a merge.

macro: causes the specified macro expansion to be inserted into the text at the indicated position.

print current date: causes the system date to be printed at the indicated position.

NOP: causes subsequent control function to be disabled (i.e. no action taken) during imaging.

NOE: causes subsequent control function to be disabled during editing but not during imaging.

replace by space: causes subsequent format effector to occupy a position on output and be printed as a space.

column backspace: cuases the active position to move to the preceding column on a row, unless it is at the first column in which case there is no effect.

zero motion: causes the normal horizontal motion after printing the subsequent character to be suppressed, causing a second subsequent character to begin printing at the same position as the first.

set tab replacement character: determines the character to be printed during a tab function.

extend line: releases the right margin, allowing text to be printed beyond the end of the line.

return to baseline: causes printer to move to first character position on print page.

return to current line: cuases printing position to move to the line at which it was prior to vertical movement.

up/down n BMUs: causes printer to change vertical print location the indicated number of BMUs.

indent n BMUs: causes printer to indent text by the number of units specified.

stop indent: terminates indentation.

return to top of page: causes the active position to return to its starting position on the imaged page.

set strikeout character: determines strike-out character to be used.

alignment tab: causes the active position to move to the column specifies as an alingment tab stop and text to ve entered to the left of the stop.

center row: causes the subsequent text on a row to be centered.

begin/end center text: delimits text to be centered.

repeat: cuases the subsequent character to be repeated a specified number of times.

right justify row: causes subsequent text up to the end of the row to be right justified.

left justify row: causes subsequent text up to the end of the row to be left justified.

Category 3 Format Effectors

begin/end faint type: delimits text to printed in faint type.

partial form feed: forces a partial form feed (useful for envelopes).

select ribbon colour: changes colour of print.

begin/end proportional printing: delimits characters which wil each occupy variable amount of space when printed.

suspend printing: causes printing to be suspended until resumed by operator.

line overstrike: causes the subsequent line to overwrite the previous line (i.e. disables the mormal vertical motion of the printer prior to printing the subsequent line).

numeric space: causes a space to be printed having the same width as a single digit in the current pitch.

unit backspace: moves the printing point left one escapement unit in proportionally spaced type styles. Treated as a backspace overwrite in monospaced type styles.

partial unit backspace: moves the printing point left zero to one escapement unit in proportionally spaced type styles.

numeric backspace: moves the printing point left a distance equivalent to the escapement value for digits in the current pitch.

Category 4 Format Effectors

positive image: causes text to be displayed in normal video.

negative image: causes text to be displayed in reverse video.

slowly blinking: causes text to be displayed at a rate less than 150 ppm.

rapidly blinking: causes text to be displayed at a rate greater than 150 ppm.

steady: causes text to be displayed non-blinking.

foreground colour: identifies the colour in which the character itself is to be imaged.

background colour: identifies the colour in which the video background.

concealed characters: causes subsequent text to not be displayed.

revealed characters: causes subsequent text to be displayed.

select intensity: determines the intensity of the video display.

2.5 Category 5 Format Effectors

The following format effectors are conceptually single characters but will require more than one byte to encode. They apply to both hard and soft copy devices.

unknown character: contains a character which is not part of the current character set. It must allow for a multi-byte sequence to encode characters from a 16-bit (or larger) character set. The character will be preserved by processing programs, but must be imaged as a single character, perhasp "?".

variable length space: a space between words which is of nonstandard width. Its width is encoded as a number of BMUs.

3. LOGICAL OPERATORS

These operators identify relationships among logical objects. The following operators are proposed:

object reference: indicates a position in the text where the designated object is referenced. Examples of objects which may be referenced are footnotes and diagrams.

start/end inclusion: delimits text which is to be included in the specified logical object. This may be used for example to identify text to be included in an index, a table of contents or a keyword list.