Date: 2010-09-12

ISO/IEC JTC 1/SC 34

Document Description and Processing Languages

Secretariat: Japan (JISC)

DOCUMENT TYPE	Disposition of Comments
TITLE	Disposition of comments for DIS 14297
SOURCE	Mr. Ningsheng Liu, Sursen Corporation (lns@sursen.com)
PROJECT NUMBER	ISO/IEC JTC 1/SC 34 N 1507 for DIS 14297
STATUS	Disposition of comments for DIS 14297 is complete based on the results of the BRM (N 1506). Please circulate to SC34 national bodies and BRM participants.
REFERENCES	
ACTION ID.	
REQUESTED ACTION	Please circulate this disposition of comments to SC34 national bodies and BRM participants.
DUE DATE	11 October 2010
Number of Pages	230
LANGUAGE USED	English
DISTRIBUTION	P, O and L Members of ISO/IEC JTC 1/SC 34 ; ISO/IEC JTC 1 Secretariat; ISO/IEC ITTF

To: Secretariat, JTC1/SC34
From: OASIS
Date: 10 August 2010
Re: Proposed Disposition Report for DIS 14297

We are pleased to transmit the enclosed Disposition Report for DIS 14297, based on the work of the OASIS UOML-X TC to revise and update our original PAS submission in response to the comments of JTC1 National Bodies.

As suggested by some commenters, the specification itself (**"UOML (Unstructured Operation Markup Language) Part 1 WS 1.0.1.pdf**") also has been revised and updated. The new current version of that document is a working draft specification from the UOML-X TC, and is visible to JTC1 and the public here: **http://www.oasis-open.org/committees/download.php/38911/UOML%20%28Unstructured%20Operation%20Markup%20Language%29%20Part%201%20WD%201.0.1.pdf**
That document includes the specification's XML Schema as Annex A.

It is the intent of OASIS to submit that set of draft revisions to JTC1 if the BRM results in a vote of approval. If, instead, the BRM colloquy results in further changes that are mutually acceptable to OASIS and to the JTC1 national bodies, then the above document will not be finally approved in OASIS in its current form; but rather, it will be revised again, and then finalized and transmitted to JTC1 after the BRM.

The submission terms of the original Explanatory Report from OASIS remain unchanged. It should be noted, in light of comments JP-02, JP-03, JP-04, JP-05 and JP-06, that OASIS will continue to follow the best practices established by JTC1, SC34, and OASIS, regarding maintenance and future versions, specifically those developed and reported to the JTC1 plenary. That is consistent with the statements in the original Explanatory Report. Thus, if DIS 14297 is approved as an IS, the terms of the submission would continue to require that any later new proposals for change, from any source, would be approved under the OASIS process, and then shared with JTC1 by submission to its approval, so that no forking of the standard occurs.

OASIS and its UOML-X Technical Committee particularly extend our thanks to the participating national bodies for their careful, thoughtful and constructive commentary, the great majority of which has been incorporated into the submission, to the benefit of the work.

Respectfully submitted,
James Bryce Clark
for OASIS

Summary

There were a total of 139 comments on the DIS 14297 spread amongst 8 different National Bodies. All of the 8 NBs who commented also voted “No” to allow 14297 to become an ISO standard. All 139 comments have been addressed.

This table summarizes the breakdown of comments and OASIS UOML-X TC responses by country.

NB	Total Number of Comments	Number of Comments Accepted by UOML-X TC	Number of Comments Partially Accepted by UOML-X TC	Number of Comments Rejected by UOML-X TC (Not Including Those Rejected Because of Substantive Change)	Number of Comments Rejected by UOML-X TC Because Change Would Require Substantive Change
CZ	6	3	2	1	0
DE	30	18	1	11	0
FR	6	6	0	0	0
GB	58	51	0	5	2
JP	10	8	0	2	0
KR	4	3	0	1	0
NL	4	4	0	0	0
NO	21	16	0	5	0
TOTALS	**139**	**109**	**3**	**25**	**2**

See here for a general comment regarding the many comments regarding the lack of an XML schema.

This table summarizes the breakdown of comments and NB response to the OASIS UOML-X TC responses.

NB	Total Number of Comments	Number of UOML-X TC Responses Accepted by NBs	Number of UOML-X TC Responses Accepted by NBs (with additional proposals/changes)	Number of UOML-X TC Responses Partially Accepted by NBs	Number of UOML-X TC Responses Partially Accepted by NBs (with additional proposals/changes)	Number of UOML-X TC Responses Rejected by NBs
CZ	6	0	3	0	2	1
DE	30	15	2	2	2	9
FR	6	2	0	0	0	4
GB	58	46	8	0	0	4
JP	10	5	1	0	2	2
KR	4	2	0	0	0	2
NL	4	4	0	0	0	0
NO	21	9	1	0	0	11
TOTALS	**139**	**83**	**15**	**2**	**6**	**33**

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
CZ-01			ge	The definition of many objects and properties is very brief and does not provide enough information about their intended purpose and use. We do not believe that the UOML submission in the current form is sufficient for building conforming implementation. There is simply too much information missing and left unexpressed. The submission is inconsistent in terms used. In section 2 the abstract document model is described. But this description is often not just a definition of constraints and properties of data model, but contains statements like "The root DOCSET will be generated automatically when the DOCBASE is created." The text of specification should use one consistent approach for describing UOML. Page 8 section 1.2 states that "This specification is the 1st part of UOML" – however there are no references to other parts in the submission. The UOML submission is thus either incomplete or contains misleading and excessive information.	None. **OASIS UOML-X TC Response** **Accepted.** The specification has been modified adding further normative and informative information. The grammar and sentence structures have been modified as needed. Note, however, the UOML XML Schema does definitively define the syntax. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd The specification has been improved drastically to meet the requirements for implementation. Regarding Parts, wording was added in the Overview. UOML may be a multipart standard in the future. 14297 is currently the first part. Part 1 specifies the basic functions to implement UOML; other Parts would be enhancements Part 1. One may choose to implement Part 1 only, or if needed, also implement other parts.	Accepted with Change	Proposal to add schema validation to the UOML conformance clause
CZ-02	1.2		ed	There are many typos in text of the submission. In many cases the text does not	None.	Accepted with	Proposal to change the currently incorrect

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				follow grammar rules of the English language and it is impossible to interpret it in an unambiguous way which is a must for an international standard. Page 8 section 1.2 the third edition of XML specification is referenced although the fourth edition is the one which is endorsed by SC34 for use in international standards produced within SC34. Page 8 section 1.2 there is a reference [xmlschema-1] pointing to W3C XML Schema version 1.1 – this version of XML schema is still in draft stages at W3C and should not be referenced from other specifications including the submission.	**OASIS UOML-X TC Response** **Accepted**. The specification has been improved grammatically. Made changes to normative references to XML. The XML Schema reference is pointing to the correct XML Schema specification already. See CZ-02 details for more information.	Change	normative reference to XML Namespaces to be 1.0, 3rd edition, and make sure all W3C references look unified (e.g. remove editors and use W3C).
CZ-03	1.2		te	Page 7 section 1.2 states that UOML is expressed “with standard XML”. However the submission does not contain any normative part with definition of XML format used by UOML. Each standard which builds on top of the XML should contain normative schema (ideally expressed using one or more languages from DSDL family – ISO/IEC 19757) – there is no such schema in the submission. Section 2.1 contains definition of objects used by UOML abstract data model. However the definition of objects and properties is insufficient. There are no datatypes specified for properties. Definitions look very informal. The submission should use some widely recognized notation for specifying objects and their properties, for example UML or IDL.	None. **OASIS UOML-X TC Response.** **Partially Accepted**. The XML language and XML Schema are both noted as normative references within the specification. And the normative UOML XML schema does include data types for the UOML objects. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd A UML based figure has been added to the specification describing the abstract document model.	Partially Accepted with Change	See JP-10 Proposal to make changes proposed for CZ-03 with additional changes of ensuring no discrepancies between figures 1, 2, 3 (including names and relationships) and adding properties on objects and their data types within the UML diagram.

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					See CZ-03 details for more information.		
CZ-04	2.3.1.2		te	Section 2.3.1.2 META object has two properties key and val. The description says that content of val property is character string. International standards should be culturally neutral and if possible they should support different writing systems and scripts. Metadata could contain texts in several languages or some explanatory annotations. Content of val property thus should support not only character strings but it should support data structure which can express things like changes in the directionality of the text (aka “BIDI”) or Ruby markup.	None. **OASIS UOML-X TC Response** **Rejected**. As specified in the UOML XML schema, both key and val are defined as xs:string, which is a normative reference type for the XML Schema specification. This string has limited restrictions. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd. Key/value pairs are an easy and approachable way to express metadata. Key/value pairs allow the UOML specification to remain simple and approachable to those who would like to understand and implement the specification.	Rejected	
CZ-05	2.3, 2.4, 2.5, 2.6		te	Section 2.3, 2.4, 2.5 and 2.6 contain definition of many properties for holding coordinates or dimensions of various objects. However nowhere is defined which measures can be used for specifying those.	None. **OASIS UOML-X TC Response** **Accepted**. Units and valid ranges will be defined for appropriate properties. Also, data ranges have been added associated with the	Accepted with Change	Multi-part proposal…. Fix the new Data Ranges clause in WD 1.0.1 to the following: 1. (No Change)

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					properties of various objects. See CZ-05 details for more information.		2. Positive, negative, or zero integer values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3). 3. Integer values are 32-bit precision; the range of integer values is as defined by xs:integer in XML Schema 1.0 Part 2. 4. Float values use double-precision; the valid range is as defined by xs:double in XML Schema 1.0 Part 2. 5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET. 6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of integer values between 0-

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
							255. 7. Valid ranges and formats for a date are as defined by xs:date in XML Schema 1.0 Part 2. Also, proposal to change the angle property in the Ellipse object to say “It is specified using a radian value of type xs:float”. Also proposal to replace inch with centimeters for the Logical Coordianate System clause in WD 1.0.1. “1/x inch” becomes “1/x 2.54 cm”. Also make a reference to the ISO standard on data types.

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
CZ-06	3		te	Section 3 defines operations on objects defined in the previous section which can be seen as definition of application interface. However the interface definition is underspecified and should use some common notation like IDL. Section 3 contains examples of XML fragments however it is not clear how these relate with operations defined. Moreover usage of XML namespace in example seems far from being uniform – some elements are namespaced, some are in no namespace. Naming conventions in presented XML fragments are not consistent – some elements use lowercase and some uppercase which is very inconvenient. There is no schema for elements used.	None. **OASIS UOML-X TC Response** **Partially Accepted**. The UOML XML Schema is a normative part of the UOML specification. All elements, data types are defined by this XSD. During the editing process, the examples have been improved to be schema-valid. However, an informative annex was added showing a detailed example of how UOML could be used. Both XML and IDL can be used to describe the API; however, XML provides a better compatibility between different versions of the standard. For example, if an additional parameter needs to be added for an instruction in the UOML standard, in the XML case, an application which understands the new standard can call the DCMS which does not understand the new standard without issue; while in the IDL case, a new interface should be added and if an application calls a DCMS which does not understand the new	Partially Accepted with Change	See GB-13

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					standard, a run-time error will be generated. See CZ-06 details for more information.		
DE-01	Overall specification		ge	Germany disagrees for the following reasons: UOML is using existing standards only for the very fundamental components, ie. XML, images formats, OpenFonts and W3C datatypes. The higher level of the UMO format does not take advantage of standards. UOML model (structure and metadata) nor the UOML viewe (layout), nor the UOML controller (operations protocol) are based on existing standards.	Re-use of existing standards (e.g. SVG, Topic Maps) for UOML model, view and controller or provide reasoning why this is not possible. **OASIS UOML-X TC Response** **Rejected**. Appropriate standards are being used to define the UOML specification. While other standards may be appropriate for UOML, the current description and use of standards in the current UOML specification is not inappropriate. There is no	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					definition conflict because of the lack of use of other standards.		
DE-02	Explanary report pg1	2	ge	"UOML specification defines an abstract model of a document" An abstract model of a document should focus on the structure, elements, features, behaviour etc. of a document. Instead UOML defines an abstract model for a document storage (docbase) and the page based graphical representation of documents. Thus UOML is more a graphic framework than a document model	Redefine purpose of UOML **OASIS UOML-X TC Response** **Partially Accepted**. UOML is a markup language to access unstructured document data, including visual aspects of a document. The purpose of UOML is specified clearly within the normative specification, particularly the UOML Overview and Abstract Document Model clauses. The model that describes the visual appearance of a document is also a document model. However, we have refined the specification to more clearly specify the purpose by adding new clauses such as the Scope, rewording clauses such as the Overview, and reformatting the entire specification for better readability.	Partially Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
DE-03	Explanary report pg 1	3	ge	"UOML provides an access interface for controlling the layout of documents…" UOML does not control the layout of documents but allows to define and manipulate a page based graphic representation of a document.	Redefine purpose of UOML **OASIS UOML-X TC Response** **Accepted**. The purpose of UOML is specified within the normative specification. However, we have refined the specification to more clearly specify the purpose by adding new clauses such as the Scope, rewording clauses such as the Overview, and reformatting the entire specification for better readability.	Accepted	
DE-04	Explanary report pg1	4	ge	"UOML defines the instruction format …" It can be doubted that XML is well suited to specify the structural and operational aspects of UOML.	Define the UOML metamodel including structural and operational aspects using a formal notation like UML. Add behavioural aspects like the required sequence of operations to the model. Add a UML -> XSD mapping as mandatory part of the standard and possibly other mappings to programming languages. **OASIS UOML-X TC Response** **Rejected**. An XML Schema	Partially Accepted with Change	See CZ-03

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					definition can be used to specify both structural and operational aspects of a standard specification.		
DE-05	Explanary report pg 4	Last on page	ge	"UOML provides detailed descriptions, explanations, examples, …"	Improve the explanations in a way that they explain the purpose of UOML and the required behaviour of a DCMS **OASIS UOML-X TC Response** **Accepted**. Detailed examples will be added as an annex showing how UOML works with respect to the object model and instructions upon the model. See DE-05 details for more information.	Partially Accepted with Change	See GB-13
DE-06	Explanary report pg 6	2	ge	"… would facilitate interoperation by providing an unified operation interface…" Provision of a well defined API does not support interoperability	Remove **OASIS UOML-X TC Response** **Rejected**. The interface specified by the normative XML schema does support interoperability. The semantics of the interface specifies the contract, but the implementer decides the mechanism by which the	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					contract is implemented. Just as SQL is an interface for data access and manipulation, there is interoperability amongst different data vendors that support SQL.		
DE-07	Strategic Characteristics pg8		ge	The feature set offered by UOML is too small and not generic enough to fulfill the promised interoperability of daily used documents or the promise given on interoperability	Remove any claim of interoperability with existing document formats or proof it. **OASIS UOML-X TC Response** **Accepted**. Provided wording in the Overview stating the possible interoperability among different software and file formats, e.g, PDF. See DE-07 details for more information.	Accepted	
DE-08	1.1		ge	In general the definitions provided in this specification are are misleading and liable to misinterpretation	1.Extend the formal definitions or 2. Provide more explanations of the meaning and 3. Provide for the use of terms a Glossary **OASIS UOML-X TC Response** **Accepted**. More terms have	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					been added to the Terminology clause and existing definitions have been clarified as needed. Also, within the specification further clarification has been added during changes based on NB comments. See DE-08 details for more information.		
DE-09	1.1	Docbase, Docset	ge	The relationship between a Docbase and the root Docset is unclear. One seems obsolete.	Revise the definitions of Docbase and Docset to make the relationship and distinction clear. **OASIS UOML-X TC Response** **Accepted**. Clarified the relationship between docbase and docset in the Terminology clause. See DE-09 details for more information.	Accepted	
DE-10	1.2	UOML is expressed with standard XML	te	If UOML is expressed in standard XML, an XML schema is required for the end user.	Provide a schema (e.g. RelaxNG) for the document structure. **OASIS UOML-X TC Response** **Accepted**. The normative XML schema has been	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					provided as part of the UOML specification. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd		
DE-11	2.1		te	The architecture shown in figure 2 is not an abstract document model	Replace "document model" by "document storage framework" **OASIS UOML-X TC Response** **Rejected**. UOML is based on an abstract document model. Note that all instances of “global data” with “global objects”, and “page data” to “page objects”.	Rejected	
DE-12	2.1.1 ff		te	We are missing definitions for the following terms that appear to have specific meanings in this specification: • sub-element • direct sub-object • indirect sub-object	We would like to see definitions for these terms **OASIS UOML-X TC Response** **Accepted**. Added a definition for sub-element in Terminology. Also removed references to direct and indirect sub-object. See DE-12 details for more information.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
DE-13	2.2-2.4		te	The section does not defines an "internal structure of a document". It defines a "document layout/presentation and storage framework"	Clarify that UOML defines the mapping of documents to a final layout, stored in a specific document data base that can be accessed via an associated API **OASIS UOML-X TC Response** **Rejected**. The document refers to the type of documents such as PDF that uses page description languages or operators to give the exact size and position of graphics objects. These describe a page, which is different from the semantics of ODF or OOXML.	Rejected	
DE-14	2.2		te	This Clause states that "Page Data may include 0 ... page(s)". We do not know document with no pages!? What about Global Date like i.e. Metadata?	Clarify how to interpret a document with no pages Clarify whether Global Data may be omitted or not **OASIS UOML-X TC Response** **Accepted**. Added clarifying wording in updated Page and Document clauses describing why 0 pages are permitted. The term “global data” has	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					been removed from the specification. The UOML XSD describes the mandatory nature of metadata and font objects. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd See DE-14 details for more information.		
DE-15	2.3.1.2		te	Key value pairs will only allow to preserve metadata in an closed environment. Existing standards as Topic Maps, RDF or just Dublin Core just be used or provide reasoning why this is not possible.	Use a standard for meta data. **OASIS UOML-X TC Response** **Rejected**. The key/value pairs used in UOML are based on the XML Schema data type xs:string, which is very flexible and with minimal limitation. Key/value pairs are an easy and approachable way to express metadata. Key/value pairs allow the UOML specification to remain simple and approachable to those who would like to understand and implement the specification.	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					A more complex metadata representation may be added in future parts of UOML.		
DE-16	2.4.1	Page	te	The properties width, height, resolution are underspecified. What are the possible units?	Reuse existing standards (e.g. CSS). **OASIS UOML-X TC Response** **Accepted**. Provided specific units for width, height and resolution (e.g. logical pixels). See DE-16 details for more information.	Accepted	
DE-17	2.4.2	Layer	te	A LAYER has multiple OBJSTREAM, an OBJSTREAM has no properties or sub-elements, but a sequence of objects. Why having multiple OBJSTREAM instead of one?	None. **OASIS UOML-X TC Response** **Rejected**. A layer holds 0 or more object streams. The reason a layer can hold many object streams is that multiple object streams may be needed to specify a related set of graphics and command objects, each of which is combined in one layer. The different related set (e.g., Security control) can be	Accepted with Change	Proposal to add wording to the object stream clause that says “A layer holds 0 or more object streams. The reason a layer can hold many object streams is that multiple object streams may be needed to specify a related set of graphics and command objects, each of which is combined in one layer. The different object streams can then be handled separately; for example, for future extensions for such functionality as security

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					handled separately.		control.”
DE-18	2.4.4		te	The page processing model does not explain how errors are to be handled. How is a conforming implementation supposed to behave if a path doesn't exist or similar exception need to be handled?	Error handling is required and needs to be described **OASIS UOML-X TC Response** **Rejected**. The UOML RET instruction specifies what occurs upon failure and error.	Rejected	
DE-19	Sec 2.5		te	Reuse existing standard for vector graphic instead of defining a new one	None. **OASIS UOML-X TC Response** **Rejected**. Existing standards for vector graphics are too complex. The UOML specification provides a simplified interface.	Rejected	
DE-20	Sec 2.5.5 & 1.3		te	Missing normative reference in 1.3 to a specification for the image file formats 'BMP' or 'TIFF', actually these are referred in the normative text	Add a normative reference for BMP and TIFF. **OASIS UOML-X TC Response** **Accepted**. Added related normative references. See DE-20 details for more information.	Accepted	
DE-21	2.5.11		te	It is unclear how the spaces property is to be interpreted. Interpretation of all properties is missing	Specify fully the data types and interpretation of all	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					properties. **OASIS UOML-X TC Response** **Accepted**. Added further clarification on the origin and the spaces properties See DE-21 details for more information.		
DE-22	2.5		te	Validate, that layout properties of existing ISO document standards can be mapped to UOML without any loss of layout/presentation related information	Add Annex with (sample) mappings. **OASIS UOML-X TC Response** **Rejected**. The ISO32000 can be mapped to UOML without loss of layout/presentation. Thus, it is not appropriate to specify the mapping in the UOML specification.	Partially Accepted	
DE-23	2.5.13.1		te	reference to the supported color model is missing	Add reference to the supported color model. **OASIS UOML-X TC Response** **Accepted**. Added normative reference to RGB in the Normative References clause.	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					See DE-23 details for more information.		
DE-24	2.5		te	Data types of all object attributes are missing	Use strong typed description of graphic elements and attributes. **OASIS UOML-X TC Response** **Rejected**. The data types are normatively specified in the UOML XML Schema XSD.	Accepted with Change	
DE-25	2.6.2.19 ff		te	Missing definition, specification, relationships, properties, values or value range of POP_GS PUSH_GS CHAR_STYLE SHADOW_WIDTH SHADOW_LEN SHADOW_DIR CLIP_AREA	None. **OASIS UOML-X TC Response** **Accepted**. All of these elements are specified within the UOML specification. A 'Valid Ranges' clause has been added to specify the general valid values of data types. See DE-25 details for more information.	Accepted	
DE-26	2.7		te	Width has no unit	Define unit. **OASIS UOML-X TC Response** **Accepted**. Units of 'logical	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					pixels' have been added for width.		
DE-27	3		te	No detailed error mechanism available	Add description of error handling rules to the standard Add operation specific error messages/codes. **OASIS UOML-X TC Response** **Rejected**. The UOML RET instruction specifies what occurs upon failure and error.	Rejected	
DE-28	3		te	No dependencies between operations available	Add dependencies, required order of operations. **OASIS UOML-X TC Response** **Accepted**. Described the order of operations in the UOML instructions clause. See DE-28 details for more information.	Accepted	
DE-29	3		ge	The UOML API is not mature. It is not precise enough to be used as a basis for implementations showing the same behaviour	Improve API as defined above. **OASIS UOML-X TC Response** **Accepted**. Given all of the	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					NB comments, the entire UOML specification, including the API, has been improved.		
DE-30	4		te	The conformance statement can't be validated	Improve the specification and defined a corresponding conformance statement that allows to validate the correct behaviour of a UOML/DCMS implementation. **OASIS UOML-X TC Response** **Accepted**. The Conformance clause has been re-worded to clarify UOML validation. See DE-30 details for more information.	Accepted	
FR-01			ge	France disapproves the PAS ISO/IEC DIS 14297 UOML for the reasons below. However, acceptance of our comments may change our position into an approval.	None **OASIS UOML-X TC Response** **Accepted**. See changes made below in response to comments FR-02, FR-03, FR-04, FR-05, FR-06.	Accepted	
FR-02			ge	The global quality of this document is very poor and JTC 1 should advice OASIS that the PAS procedure be used with more accuracy.	None **OASIS UOML-X TC Response** **Accepted**. Restructured and reformatted the specification to provide better quality. The	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					specification has been revised editorially and technical details have been strengthened.		
FR-03			ge, te	1) This document is very incomplete on many aspects: * Almost no units or data types are provided (width, height of page? resolution? tl and br coordinates, etc.) * No information is provided for allowed values (what does it mean to have a width with a negative value)	None **OASIS UOML-X TC Response** **Accepted**. The specification has been updated in various clauses to clarify units and proper values, including a Data Ranges and Logical Coordinate System and Units clause. Data types are defined, however, in the normative XML Schema definition. See FR-03 details for more information.	Rejected	
FR-04			ge, te	2) It is very hard to understand when the API starts and where the Format stops * What is the definition of a sub-element? * What is a sub-object? * What is the relationship between those two?	None **OASIS UOML-X TC Response** **Accepted**. Added definition of sub-element to Definitions clause. Also added informative note in the Abstract Document Model and UOML Instructions clause describing the differences between sub-element and sub-object. See FR-04 details for more information.	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
FR-05			ge, te	3) The RFC 2119 (about the correct use of SHOULD MAY MUST, etc.) is not used at all: It should be used * What are the levels of conformance?	None **OASIS UOML-X TC Response** **Accepted**. Added a Conformance clause to the specification. See FR-05 details for more information.	Rejected	
FR-06			ge, te	4) The document is entitled First part: * Why the first part? * What are the other parts about? * Why is it split into parts? * Are we voting on a multipart standard, by seeing only the first part?	None **OASIS UOML-X TC Response** **Accepted**. Added wording in the Overview clause. UOML may be a multipart standard in the future. 14297 is currently the first part. Part 1 specifies the basic functions to implement UOML; other Parts would be enhancements to Part 1. One may choose to implement Part 1 only, or if needed, also implement other parts. See FR-06 details for more information.	Rejected	
GB-01			ge	The standard specifies an abstract interface onto a DCMS (DoCbase Management System). There is nothing wrong in principle with abstract interfaces, and in general no reason why they should not even be candidates for standardization. Within	None **OASIS UOML-X TC Response** **Rejected**. Metadata in implementations of many software applications, including standards (e.g.,	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				SC 34 the (non-standardized) SAX abstract interface is widely used; and more widely the (ISO/IEC) SQL standard could be seen as an abstract interface in its query language part, to which UOML compares itself. However, in both these examples the interface is layered on top of a very well defined concrete model: in SAX's case the infoset items of XML documents, in SQL's case storage systems governed by the Relational Model. UOML is layered on top of the (unspecified) DCMS, which is itself an unspecified abstraction of collections of documents. The missing information is how DCMS represents various kinds of documents. Without this knowledge it is impossible to implement UOML reliably so that implementations interoperate. To take a concrete example, consider Clause 2.3.1 (Metadata). We are told "[g]eneral information, such as the document's title, author, creation and modification date, is called metadata." and that the METALIST operators yields "1 to multiple META" items where each META item is a key/value pair. Faced with implementing this an implementor must know, for each document type, what the "metadata" is. We are merely given the examples ("document's title, author, creation and modification date") -- but what else is to	Common Language Infrastructure) have implementation-defined behaviour. In the case of UOML, both Metadata and Font need not be uniform across implementations in order to achieve the main crux of interoperability which comes within the Page Data of a document.		

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				be included? application settings? IDs? bibliographic data? comments? And how do items such as RDF triples get represented as pairs? To be interoperable we need to have a precise specification of what metadata is for the document types (e.g. OOXML, ODF, etc.) which are envisaged being handled by the system -- otherwise there is virtually no likelihood two systems would interoperate. It might be argued that the choice of what to treat as metadata is up to the implementor, but in that case UOML is contributing very little. In particular it is not bringing the one thing we require of it: standardized behaviour.			
GB-02			ge	It is stated in the Explanatory Report accompanying the specification that: "There are two parts of software involved in the use of the UOML standard, one is the application software that consumes or displays documents; the other is a DCMS (DoCbase Management System)". How is a DCMS to be implemented? What specifications and IPR regimes govern DCMS implementations?	Provide a reference model for an implementation of UOML that describes the necessary software components of the implementation. If a DCMS is a necessary component of an implementation, provide examples of how this component might be implemented (e.g. using a file system, a relational database management system, an XML database management system). **OASIS UOML-X TC**	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Response **Accepted**. The DCMS is the layer that stores documents and executes UOML instructions. The implementation of a DCMS is outside the scope of this standard as long as it implements the semantics of the instructions correctly. A reference model of a DCMS or an application that displays documents from a DCMS is outside the scope of the current specification. That said, a simple example of a possible DCMS implementation has been provided in this disposition of comments. See GB-02 details for more information.		
GB-03			ge	The specification provides no information on the business or use case that UOML is intended to meet. It would appear that all existing implementations of this specification have been in connection with use of a particular DCMS implementation by the company Sursen. It does not seem to us that this constitutes a sufficient case for proposing an International Standard. We have been unable to identify any use cases for this specification within the United Kingdom.	Provide an adequate business case by adding appropriate text to Clause 1.2. **OASIS UOML-X TC Response** **Accepted**. Provided a summarized business case in the Overview sub-clause. Also provided a more detailed business case in this disposition of comments. See GB-03 details for more	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					information.		
GB-04			ge	The specification does not include a clear statement its Scope. There is an informal description in Clause 1.2, but this doesn't provide what is needed.	Add a clear statement of Scope, possibly to Clause 1.2, in accordance with the requirements of JTC 1 standards. **OASIS UOML-X TC Response** **Accepted**. A Scope clause has been added. See GB-04 details for more information.	Accepted	
GB-05			ge	The specification is labelled as being Part 1 of, presumably, a multi-Part specification. Yet no information is provided about any other Parts, or about the dependencies between them.	Add text to Clause 1.2 that explains the anticipated relationships between this specification and other Parts. **OASIS UOML-X TC Response** **Accepted**. Added wording in the Overview clause. UOML may be a multipart standard in the future. 14297 is currently the first part. Part 1 specifies the basic functions to implement UOML; other Parts would be enhancements Part 1. One may choose to implement Part 1 only, or if needed, also implement other parts. See GB-05 details for more detailed information.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
GB-06			ge	The specification makes normative reference to several standards and reference specifications, and also makes a non-normative reference to ISO/IEC FDIS 32000. However there is no explanation of how this specification relates to ISO/IEC FDIS 32000, nor to a number of other important standards used in layout-oriented representations of documents (e.g. ISO 10180 SPDL, XSL-FO, SVG).	As part of the business case for this specification, provide an explanation in Clause 1.2 of how UOML relates to existing standard page description and page formatting languages. **OASIS UOML-X TC Response** **Accepted**. Added explanation in the Overview of how PDF plays a role with UOML. See GB-06 details for more information.	Rejected	
GB-07			ge	The specification as a whole is lacking in detail, making it frequently difficult to interpret what is meant. The Clause-specific comments below cover the worst cases that have been found, but there is a general sense in which this specification has not been editing for clarity. While it is no doubt self-evident to the authors what is meant, given their familiarity with the concepts and their experience with existing implementations, it should not be assumed that someone unfamiliar with the technology will start with the same level of conceptual understanding.	The entire text needs to be edited for sense, to make it intelligible to technically literate implementers who are nonetheless unfamiliar with the specific concepts that underpin this specification. Much could be achieved by providing a comprehensive glossary, by expanding the existing Clause 1.1 (Terminology), which may reduce the extent to which text more generally needs to be edited, and by providing a reference model of an implementation. **OASIS UOML-X TC Response**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Accepted. The text will be edited for grammatical clarity and definitions will be added to the Terminology clause as necessary. See GB-07 details for more information.		
GB-08			ge	The specification claims to define a language that uses XML as its representation. However there is no normative specification of the XML syntax for UOML. The abstract model is not formally defined, relying on narrative text and two Figures (1 and 2).	Add formal text to specify the abstract model and a reference XML syntax for UOML. The abstract model might, for example, be specified in EBNF or UML, while the XML model should be provided in the form of a RELAX NG schema. **OASIS UOML-X TC Response** **Rejected**. The UOML XML schema (XSD) is the normative reference for the description of the functional elements of UOML. RELAX NG could have been used instead of the W3C XML Schema definition. RELAX NG may be considered in a future version of the standard, but since W3C is an acceptable and known standard, for now it will remain.	Accepted with Change	Proposal to change informative schema of UOML WD 1.0.1, lines 1725 and 1746, from xsd: to xs:. Also, propose that informative schema replace the external normative schema since the informative schema contains element and type information (xs:base64binary) for EMBEDFONT and IMAGE. In other words, change the XML Schema for EMBEDFONT and IMAGE to reflect the content model of the working draft prose in appropriate sections. See GB-13 See JP-10
GB-09	1.1		ge	In general the definitions provided in this specification are too terse and therefore	Either extend the formal definitions or provide longer	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				liable to misinterpretation.	explanations of the meaning and use of terms in a Glossary. **OASIS UOML-X TC Response** **Accepted**. More detail will be added to the Terminology clause as needed. Also a detailed example on the use of UOML and its features will be added as an Appendix. See GB-09 details for more information.		
GB-10	1.1	Docbase, Docset	te	The relationship between a Docbase and the root Docset is unclear. The term Docset is sometimes capitalised, sometimes not. If there is no distinction, the same form should be used throughout.	Revise the definitions of Docbase and Docset to make the distinction clear. Apply consistent spelling of key terms throughout the specification. **OASIS UOML-X TC Response** **Accepted**. Added further wording to explain the relationship between Docbase and Docset. Also ensured consistent use and spelling within the specification. See GB-10 details for more information.	Accepted	
GB-11	1.1	Layer	te	What is being defined here is both 'Page' and 'Layer'.	Create separate definitions for Page and Layer.		

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					OASIS UOML-X TC Response **Accepted**. Refined the definition of Layer in the Terminology clause. See GB-11 details for more information.		
GB-12	1.1	Path	te	It is confusing to have a single definition for the term 'Path', but in the definition include the definition of another term 'path' that can easily be confused with the term being defined – particularly as the term is also used in capitalised form 'PATH'. The purpose of a Path is not clearly explained either in the definition here or in the body of the specification (Clause 2.5.10).	Create separate definitions for 'Path' and 'path', and if possible use two terms with a greater distinction in their spelling. **OASIS UOML-X TC Response** **Accepted**. Changed part of the definition of Path to an informative note specifying the distinction between the two terms in question. Modified some wording in the PATH element to clarify that it is the actual UOML object and not conceptual. See GB-12 details for more information.	Accepted	
GB-13	1.2	Paragraph 1	ge	It is claimed that the specification uses XML syntax, but there are very few examples of how to express UOML in XML. The Explanatory Report states that the reference schema provided is non-normative. This presumably means that the few XML examples in the text of the	Provide more Examples in XML syntax throughout the specification. Make it clear that Examples are non-normative. **OASIS UOML-X TC**	Accepted with Change	See GB-08 Proposal that namespace prefixes (e.g., x:) be added to the outer elements of the examples in the newly added Annex B of UOML

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				specification are non-normative. See Example in Clause 2.6.1 and several Examples in Clause 3. Such Examples are helpful for understanding the specification and could usefully be provided in many other Clauses, but it should be made clear that these Examples are non-normative.	**Response** **Accepted**. The provided UOML schema is indeed normative and a detailed example will be provided in the Appendix. All examples within the specification text, not in appendices, will be marked as non-normative. See GB-13 details for more information.		WD 1.0.1, starting at line 2159
GB-14	1.2	Paragraph 2	ed	This paragraph contains background information that should probably not be normative. It should either be moved to a Foreword or made into a Note.	Move non-normative text to Foreword or to a Note. **OASIS UOML-X TC Response** **Accepted**. The Overview is non-normative by default. The entire clause is non-normative. That has been made more clear in the updated specification.	Accepted	
GB-15	1.3		te	Normative reference is made to W3C XML Schema Parts 1 and 2, but there is no use of these references in the text.	Either make explicit use of these normative references or remove them. **OASIS UOML-X TC Response** **Rejected**. The UOML	Accepted with Change	See GB-08 See GB-13

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					element definition is normatively defined via an XML-Schema. See Annex A in the updated specification for a copy of the normatively defined XSD file.		
GB-16	1.3		te	There is no normative reference to a specification for the image file formats 'BMP' or 'TIFF', yet these are referred to in normative text (in Clause 2.5.5)	Add a normative references for BMP and TIFF. **OASIS UOML-X TC Response** **Accepted**. Added related normative references in the Normative References clause. See GB-16 details for more information.	Accepted	
GB-17	1.3		te	There is no normative reference to the definition of the "rgb" colour space referred to frequently in the text.	Add a suitable normative reference. **OASIS UOML-X TC Response** **Accepted**. Added a normative reference to RGB. See GB-17 details for more information.	Accepted	
GB-18	1.4		ed	Non-normative reference is made to ISO/IEC FDIS 32000, but there is no use of this reference in the text.	Either provide text to explain why ISO/IEC FDIS 32000 is included in the non-normative references, or remove it.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					OASIS UOML-X TC Response **Rejected**. Duplicate comment as GB-06		
GB-19	2		te	The data types of properties are generally not specified in Clause 2. In a few cases the data type is specified somewhat vaguely, e.g. "character string", "floating point number".	Specify data types for all properties. **OASIS UOML-X TC Response** **Accepted**. All data types are specified in the normative UOML XML schema supplied as an XSD with this specification. Annex A was also added for further clarity.	Accepted with Change	See GB-08 See GB-13
GB-20	2 and 3		te	The cardinalities of sub-elements and sub-objects are sometimes specified in narrative text but are not specified in the formal specifications of each object or element. In some cases the cardinality is completely missing.	Add cardinalities to formal specifications where missing. **OASIS UOML-X TC Response** **Accepted**. Cardinalities are specified in the normative schema with “minOccurs” and “maxOccurs”. A UML diagram has been added in the clause describing the abstract document model showing cardinalities as well. General data ranges are also specified in the Data Ranges clause. See GB-20 details for more	Accepted with Change	See GB-08 See GB-13

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					information.		
GB-21	2.1 and 2.2		ed	It is unclear whether Figures 1 and 2 are intended to be normative. They appear to contain information that is not formally stated in the text. The figure captions do not make it clear that they are illustrating different parts of the same model.	Make it clear that Figures are non-normative, and ensure that all information in Figures is formally stated in normative text. Improve the captions by inserting the words 'Part of the' before 'UOML' in each case. **OASIS UOML-X TC Response** **Accepted**. The figures are intended to be normative and the terms used in the figures are changed to be the same as in normative text.	Accepted with Change	See GB-21
GB-22	2.1.1 and following		te	There are no definitions for the following terms that appear to have specific meanings in this specification: • sub-element • direct sub-object • indirect sub-object (Note that "sub-element" is not defined in XML 1.0.)	Provide definitions for these terms. **OASIS UOML-X TC Response** **Accepted**. Added a definition for sub-element in Terminology. Also removed references to direct and indirect sub-object. See GB-22 details for more information.	Accepted	
GB-23	2.1.3, 2.2		ed	The term "Document Global Data" is used several times, but the term used	Either adopt a consistent	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				previously is "Global Data", assuming that these are the same thing.	term, or define the distinction between "Global Data" and "Document Global Data". **OASIS UOML-X TC Response** **Accepted**. The one instance of Global Data should have been Document Global Information. This has been remedied. Also, a definition of document global object as been added to the Terminology clause. [document global object: A document global object may include a fontlist, fontmap and/or embedfont. These global objects may be used and referenced globally throughout the entire document.]		
GB-24	2.2	Paragraph 2	ed	The sentence "Document Global Data can be used globalize among the document, it includes Metadata and Font" is poorly drafted and could therefore be subject to inconsistent interpretation. The term "Font" is used here and in several other places, but the terms "Font Data" and "Font Definition" are also used. It is unclear if these are synonymous. If they are, a consistent	Clarify the sentence using grammatical English. Adopt consistent terminology for "Font". **OASIS UOML-X TC Response** **Accepted**. Rewrote stated sentence to be more grammatically correct and	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				term should be used.	removed instances of “Font Definition” and “Font Data” See GB-24 details for more information.		
GB-25	2.2		te	This Clause provides only a vague outline of the architecture of a document. No detail is given. It is not possible to implement this, which is the key part of the specification. Examples of the kind of detail that is missing include: • the units of measurement that are applicable to specifying the dimensions and positions of objects (there is no mention of measurement units anywhere in the specification), or how such units might be specified • the relationships between layers on a page • the relationships between "command objects" and the page / layer / graphical objects whose properties they modify	Provide a complete overview of the document object model and the document processing model. **OASIS UOML-X TC Response** **Accepted**. Further detail is provided in other normative clauses, including Terminology. However, more detail on units of measurements has been provided.	Accepted	
GB-26	2.2		te	This Clause states that "Page Data may include 0 ... page(s)". Is it meaningful to have a document with no pages? Is it valid for a document to have no Global Data, i.e. no Metadata and no Font Data? It is not specified whether these can be omitted or are required in all documents.	Clarify how to interpret a document with no pages, or change the cardinality to require at least one page in a document. Clarify whether Global Data may be omitted or not.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					OASIS UOML-X TC Response **Accepted**. Added clarifying wording in the Page sub-clause describing why 0 pages are permitted and that global objects are not required in a document. See GB-26 details for more information.		
GB-27	2.3.1.2		te	The property key is under-specified, particularly in terms of value constraints. For example, are two META objects allowed to have the same value of key?	Specify fully the data type for this property, including any string-length and uniqueness constraints that may apply. **OASIS UOML-X TC Response** **Accepted**. Valid values will be described throughout the entire document. Also, wording has been added to state that the key/value properties are not necessarily unique for a given instance of the element. See GB-27 details for more information.	Accepted	
GB-28	2.3.2		ed	The sentence "EMBEDFONT, FONTMAP, FONTLIST in the order give above, the previous one is the sub-object of the latter one, and can be generated by UOML's INSERT	Re-write the sentence in grammatical English. **OASIS UOML-X TC**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				instruction" is unclear and liable to a variety of interpretations.	**Response** **Accepted**. The sentence was rewritten for better clarity. See GB-28 details for more information.		
GB-29	2.4.2, 2.4.4		te	The relationship between layers on the same page is unclear. A Layer apparently has no properties, but it possibly needs at least one property to specify whether a layer is opaque or transparent. It may be that all such properties are determined by the objects within the Layer, but at least some objects (e.g. Text) do not define opacity. There is no way in which the order of Layers on a page can be specified, other than implicitly in their order within an XML representation. There is therefore no certainty as to how to interpret the first step in Clause 2.4.4: "Repeat the following step[s] from the first layer to the last layer"	Add text to explain the ordering and visual properties of Layers, and how multiple Layers inter-relate on a single page. **OASIS UOML-X TC Response** **Accepted**. Added text in the Layer clause to explain the relationship between same page layers. See GB-29 details for more information.	Accepted with Change	A multi-part proposal that a definition of position number be added to the Terminology clause (Integer starting at 0 to some implementation-defined maximum, which defines a sequence of objects), the page rendering algorithm loop through the object stream by position number (add "by position number" to the end of lines 318, 320, 321 in WD 1.0.1), a clarification that position numbers are incremented on an insert when inserting an object in the middle of a sequence (In INSERT, wording considered may be: "pos: int value, starting from 0, representing the insert location. The object shall be inserted before the object at pos. This property is optional. If this property is not used, insert

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
							after the last sub-object. If pos is greater than or equal to the number of items in the sequence then the insertion point is implementation-defined. After the insertion, the position numbers of all items after the inserted item are increased by one." In DELETE, "After a deletion, the position numbers of all items are after the deleted item is implementation-defined", a definition of implementation-defined to Terminology, a clarification be added regarding the result of building layers upon each other.
GB-30	2.4.4, 2.5 and 2.6		te	Graphics objects in other page description languages include many more properties than are listed for each of the objects specified in 2.5. It would appear that in UOML these properties are not specified on the graphics object itself, but are specified by a "command object". What is not clear is how to determine which command object influences which graphic object. The page processing model in 2.4.4 could be interpreted to imply either that a command object affects all graphics objects in the same OBJSTREAM, or that it affects only the graphics object	Add text to Clause 2.4.4, and elsewhere as necessary, to explain how a set of objects in an OBJSTREAM are rendered based upon the objects that it contains. Change the heading of Clause 2.4.4 from "Page processing models" to "Page processing model" – there is only one model given. **OASIS UOML-X TC Response**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				that it follows (precedes?).	**Accepted**. Added a Graphics State clause explaining how a command object affects the graphics state. See GB-30 details for more information.		
GB-31	2.4.4		te	The page processing model does not explain how errors are to be handled. How is a conforming implementation supposed to behave if a path doesn't exist, a value is out of range, etc?	Specify error handling requirements. **OASIS UOML-X TC Response** **Accepted**. The UOML operations do specify error handling requirements. For example, the OPEN operation specifies what the return value shall be upon success or failure. However, the specification has been refined with additional wording and examples.	Accepted	
GB-32	2.5 and sub-clauses		ed	The coordinate system being used to define Graphics Objects is inconsistently and incompletely defined. The terms "position", "point" and "coordinate" appear to be used interchangeably. No units are defined, nor any method to specify them. The use of radians as the unit of measurement for angles is idiosyncratic and may cause interoperability problems with other page description techniques,	Clarify the definition of the coordinate system being used. Make terminology consistent throughout. Provide clear justification for adopting radians rather than degrees as the unit of angle measurement. **OASIS UOML-X TC**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				such as PDF, XSL-FO and SVG, which use degrees.	**Response** **Accepted**. Added clarifying wording in a Logical Coordinate System and Units clause to describe the logical coordinate system being used. Also discussion regarding resolution and radians were added. See GB-32 details for more information.		
GB-33	2.5.1	Properties	te	The relationship between an ellipse arc's clockwise and angle properties is unclear, making it appear that contradictory properties are possible (one "clockwise", the other "anti-clockwise").	Clarify the definition of the properties of an ellipse arc, possibly by including a suitable Figure. **OASIS UOML-X TC Response** **Accepted**. Valid values for clockwise and angle will be provided, stating boundaries between the two values. The value of an angle is always positive; a negative value is not allowed. See GB-33 details for more information.	Accepted	
GB-34	2.5.5	Note	te	The Note refers to a property imgpath, but no such property is specified for the object IMAGE, which has a property path.	Correct the specification of the property, or correct the Note. **OASIS UOML-X TC**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Response **Accepted**. Changed 'imgpath' to 'path'.		
GB-35	2.5.8	Properties	te	Two radius properties are specified for the object ROUNDRECT. In other page descriptions languages rounded corners are generally circular, so only one radius is specified. If elliptical corners are being specified here, this should be made explicit. If a circular corner is required, it is unclear whether both properties xr and yr should be specified with the same value, or one of the properties should be omitted.	Clarify the definition of the properties xr and yr. **OASIS UOML-X TC Response** **Accepted**. Added wording stating that the corners are elliptical in nature. See GB-35 details for more information.	Accepted	
GB-36	2.5.9		te	It is unclear how a SUBPATH object would be expressed. An Example would be helpful.	Improve the specification of SUBPATH, possibly by including an Example in XML. **OASIS UOML-X TC Response** **Accepted**. An example will be provided showing the expression of SUBPATH. See GB-36 details for more information.	Accepted	
GB-37	2.5.10		te	It is unclear how a number of graphics objects are combined together to form a PATH. This needs further explanation with illustrative Examples.	Provide further explanation of the purpose of a PATH object and how it is to be constructed and rendered.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					OASIS UOML-X TC Response **Accepted**. An example will be provided showing the expression of PATH. See GB-37 details for more information.		
GB-38	2.5.11	Properties	te	It is unclear how the spaces property is to be interpreted. How is the "origin" of a character defined? How is distance specified? Can distances be negative? Does the number of comma-separated values have to be the same as the number of characters in the string? Do these values override the widths of the characters as specified by the font used? How are these values affected by the action of an applicable command object (which may specify font, size, weight, other transformations)?	Specify fully the data types and interpretation of all properties. **OASIS UOML-X TC Response** **Accepted**. Added further clarification on the origin and the spaces properties. See GB-38 details for more information.	Accepted	
GB-39	2.5.12		ed	A form of BNF notation is used, but there is no normative reference to the specification of this notation. The first sentence includes the word 'path' with no capital letter, but 'Path' is probably intended here.	Add a normative reference in 1.1 to the specification of the form of BNF used here. Replace 'path' with 'Path' – or better still, find an alternative term to 'Path' and use throughout. **OASIS UOML-X TC Response** **Accepted**. Added a normative reference to BNF	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					and changed “path” to “Path”. See GB-39 details for more information.		
GB-40	2.5.13		ed	The inclusion of this Clause at this point is confusing, since its contents is used only in Clause 2.6.	Move this Clause to the end of Clause 2.6. **OASIS UOML-X TC Response** **Accepted**. Moved the current 2.5.13 clause as the final subclause of the Command Objects clause.	Accepted	
GB-41	2.5.13.1		te	The meaning of 'red', 'green', 'blue' and 'alpha' is not clear, because the colour standard or reference specification is not stated.	Either provide a reference to a standard or specification in which 'red', 'green', 'blue' and 'alpha' are defined, or define them here. **OASIS UOML-X TC Response** **Accepted**. Added a normative reference to RGB. See GB-41 details for more information.	Accepted	
GB-42	2.6.2	Heading	ed	The heading of this Clause is unclear. It should probably be "CMD name and value properties and sub-elements".	Improve the clarity of the heading. **OASIS UOML-X TC Response** **Accepted**. Renamed the	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					heading of the clause to "Values for CMD's 'name' property".		
GB-43	2.6.2.6		te	It is unclear whether "line width" means "line length" or "line thickness". If (as we suspect) the latter, the name LINE_THICKNESS would be clearer.	Clarify the meaning of "line width", and if appropriate rename the object LINE_THICKNESS. **OASIS UOML-X TC Response** **Rejected**. The ISO/IEC PDF specification also uses the term "line width".	Accepted	
GB-44	2.6.2.12		te	The specifications of the various raster operations are not sufficiently clear. Is the Note normative? If so, it should not be a Note.	Provide clear specifications of the syntax of raster operations as normative text. **OASIS UOML-X TC Response** **Accepted**. Renamed 'Note' to 'Details' to imply normative text.	Accepted	
GB-45	2.6.2.13		ed	The choice of values and the explanation of what each of them means is unclear. It might be preferable to adopt conventions adopted in other languages, such as XSL-FO, e.g. TOP_BOTTOM, BOTTOM_TOP, LEFT_RIGHT and RIGHT_LEFT. Also, the Semantics definition appears to be wrong: surely the direction is from the beginning to the end, not from the end to	Improve the specification of TEXT_DIR. **OASIS UOML-X TC Response** **Rejected**. Although adding additional property values TOP_BOTTOM, BOTTOM_TOP,	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				the beginning.	LEFT_RIGHT, RIGHT_LEFT may be reasonable and easier to understand by some, adding such an instruction would break existing implementations. The comment may be reasonable, but it must be balanced against a breaking change. And, the current property values are also reasonable. Adding these property values as semantically equivalent to the current values may be considered in the future.		
GB-46	2.6.2.14		ed	It is unclear how this Clause relates to Clause 2.6.2.13. How is the "direction" of a character defined?	Clarify the meaning of "character direction" **OASIS UOML-X TC Response** **Accepted**. Clarified the meaning of character direction. The direction is from the bottom to the top in the font coordinate system. See GB-46 details for more information.	Accepted	
GB-47	2.6.2.16	Properties	te	The property v1 of the command object CHAR_SLANT is inadequately specified. It is not clear whether "right" always means the same thing, regardless of reading direction, or whether it actually	Clarify the meaning of "left" and "right" with respect to slant. **OASIS UOML-X TC**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				means "in the direction of reading".	**Response** **Accepted**. Clarified that slants are regardless of reading direction. The direction of a character's top and bottom is the same as defined in the font coordinate system. See GB-47 details for more information.		
GB-48	2.6.2.18		te	It is unclear how to interpret a decimal value between 0 and 1 as a character weight.	Specify the algorithm for determining the function to apply to a character's outline based upon a decimal value between 0 and 1. **OASIS UOML-X TC Response** **Accepted**. The algorithm will be specified in the CHAR_WEIGHT clause. See GB-48 details for more information.	Accepted	
GB-49	2.6.2.19	Properties	te	How are the properties of CHAR_STYLE to be interpreted? Is this implementation-dependent? If so, this needs to be stated. If not, the allowed values of this property need to be enumerated and defined.	Clarify whether the values and meanings of the property v1 are implementation-dependent or not. **OASIS UOML-X TC Response** **Accepted**. Specified normatively the details on the	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					three properties of CHAR_STYLE. A detailed algorithm has been added to the specification. See GB-49 details for more information.		
GB-50	2.6.2.20	Semantics	te	It is unclear whether this command applies to each character individually within a TEXT object, or to the TEXT object as a whole.	Clarify the applicability of this command object. **OASIS UOML-X TC Response** **Accepted**. Described that the command applies to each character individually within a TEXT object. See GB-50 details for more information.	Accepted	
GB-51	2.6.2.24, 2.6.2.25		te	The command objects PUSH_GS and POP_GS make reference to a "stack", but no stack is defined as part of the page processing model in Clause 2.4.4.	Define the term "stack" and include it in the page processing model in Clause 2.4.4. **OASIS UOML-X TC Response** **Accepted**. Defined the term 'graphics state stack' in the Terminology clause. See GB-51 details for more	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					information.		
GB-52	2.6.2.26 – 2.6.2.30		te	It is unclear how to interpret the way in which these command objects specify a shadow to be applied to a character. The object SHADOW_WIDTH appears to refer to the thickness of the outline of a shadow. If so, this should be clarified. It would help if the name were clearer. The object SHADOW_LEN appears to refer to the displacement of the shadow with reference to the character, and presumably the direction of displacement is given by the object SHADOW_DIR. If so, this should be clarified. The range of values specified for SHADOW_DIR look somewhat restrictive compared with what is possible in applying shadow to text in many graphics design systems. Why is the direction not specified as an angle (in radians, of course)? What is meant by adjusting the coordinates of a character when the direction of shadow is to the left and top? Should there not be equivalent command objects for adjusting the coordinates of a character when the shadow is oriented in some other direction? If the command object SHADOW_NEG is applied, what determines the thickness of the border of the character? Is this SHADOW_WIDTH?	Clarify the specification of command objects associated with applying shadows to characters. **OASIS UOML-X TC Response** **Accepted**. Added further description, figures and clarification to SHADOW_WIDTH, SHADOW_LEN, SHADOW_NEG and SHADOW_DIR. See GB-52 details for more information.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
GB-53	2.6.2.31		te	It is unclear how a CLIP_AREA command object is to be applied to the OBJSTREAM that contains it. Is the Path specified by this object relative to the Page or relative to some coordinate origin for the OBJSTREAM?	Clarify how a CLIP_AREA is applied to an OBJSTREAM. **OASIS UOML-X TC Response** **Accepted**. Clarified how a CLIP_AREA is applied to an OBJSTREAM in the CLIP_AREA clause. See GB-53 details for more information.	Accepted	
GB-54	2.6.2.32		te	It is unclear how a system is supposed to interpret a character string representing either an encoding / character set or a font. How is a consuming system intended to handle a string that it doesn't recognise?	Add references to the encoding / character set and font identification schemes that define the value domains of these font properties. **OASIS UOML-X TC Response** **Accepted**. There is a normative reference to the Open Font Format ISO standard in the Normative References clause. Also added a normative reference to INANA charsets.	Accepted	
GB-55	2.6.2.33 –2.6.2.35		ed	The term "outline border width" is unclear and should be changed to "outline border thickness". The term "outline width" is unclear and should be changed to "outline thickness".	Improve clarity by replacing "width" with "thickness". **OASIS UOML-X TC Response**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Rejected. Although OUTLINE_BORDER_THICKNESS may be easier to understand by some, adding such an instruction would break existing implementations. The comment may be reasonable, but it must be balanced against a breaking change. Furthermore, the term width is an official term from ISO 32000 (PDF). OUTLINE_WIDTH is more suitable than OUTLINE_THICKNESS. However, OUTLINE_THICKNESS as semantically equivalent to OUTLINE_WIDTH may be considered in the future.		
GB-56	3.4	Properties	te	The property value GET_PAGE_BMP appears to return an object that is not defined by this specification. If a page bitmap is simply a page containing a single graphics object in BMP format, this should be explained.	Either "page bitmap" has to be added to the UOML abstract model, or it should be explained how it fits into the abstract model, or this value should be removed. **OASIS UOML-X TC Response** **Accepted**. Defined page bitmap in the Terminology	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					clause. See GB-56 details for more information.		
GB-57	3.8	Semantics	ed	The second paragraph appears to duplicate what is contained under 'Sub-element'. There is reference to "this version". What does this refer to? This version of the specification?	Clarify this Clause. **OASIS UOML-X TC Response** **Accepted**. Removed the duplicate wording in the Semantics of SYSTEM. Also, removed the word "version" and replaced it with "Part". Thus it is "within this Part of the UOML specification"	Accepted	
GB-58	3.10.4		te	While 1.3 contains a reference to the XML Schema Part 2: Datatypes Second Edition, there is no statement that the data type referred to in this Clause (or any other) is to be taken from that W3C specification, or of which sub-formats of XML dates are valid. If one were to specify a date such as 0101 to indicate first of January, without specifying the year, is this valid?	Clarify the specification of data types, and what restrictions apply to their use. **OASIS UOML-X TC Response** **Accepted**. While the data type of DATE is specified normatively in the UOML XML Schema, valid values for DATE will be described. See GB-58 details for more information.	Accepted with Change	Proposal that valid date ranges should reference XML Schema Part 2 (instead of ISO 8601) in 4.14 of the revised WD 1.0.1 specification. And fix 4.13.6 of the revised specification to make 'datetime' camel-case 'dateTime'.
JP-01	All		ge	It is not clear what the scope of this standard is and what is intended to be standardized. As demonstrated by the sheer number of undefined terms as well	None. **OASIS UOML-X TC Response**	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				as the poor quality of BNFs, schemas, and API descriptions, the quality of this standard is extremely low.	**Accepted**. Added a Scope clause to the specification and have improved the overall specification for greater clarity. See JP-01 details for more information.		
JP-02	Explanatory Report		ge	0) Changes required by member bodies Given the quality of the submitted document, Japan believes that MB comments cannot be satisfactorily addressed without changing the submitted document substantially. Japan requests that OASIS publish a revision of OASIS UOML in reply to the required changes.	None. **OASIS UOML-X TC Response** **Accepted**. All NB comments have been addressed and the specification has been changed, as needed, to meet the results of the addressing of the comments. A revised specification will be made available.	Rejected	
JP-03	Explanatory Report		ge	1) Defect reports SC34 member bodies have the right to submit defect reports on standards maintained by SC34. Will such defect reports be handled as specified in 15.4 "Correction of Defects" in the JTC1 directives? In particular, will the OASIS UOML TC create errata in reply to such defect reports? Follow the best practises being established by JTC1, SC34, and OASIS.	None. **OASIS UOML-X TC Response** **Accepted**. If standardized within ISO, OASIS will handle defect reports as appropriate.	Accepted	
JP-04	Explanat		ge	2) OASIS errata and SC34 DCORs	None.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
	ory Report			When OASIS publishes UOML errata, SC34 has to conduct a DCOR ballot before an equivalent COR is published. Suppose that member bodies submit comments as part of their DCOR votes. How will such comments be addressed? Will the original OASIS errata be changed or more errata be created? Follow the best practises being established by JTC1, SC34, and OASIS.	**OASIS UOML-X TC Response** **Accepted**. If standardized within ISO, OASIS will handle errata reports as appropriate.		
JP-05	Explanatory Report		ge	3) Proposals for amendments or revisions SC34 member bodies have the right to submit new work item proposals for amendments or revisions. Do the explanatory report and OASIS liaison policy mean that such proposals will be handled by the OASIS UOML TC? Furthermore, the SC34 plenary can endorse project subdivision for amendments. Do the explanatory report and OASIS liaison policy mean that the SC34 plenary is required to reach consensus with the UOML TC in advance? Follow the best practises being established by JTC1, SC34, and OASIS.	None. **OASIS UOML-X TC Response** **Accepted**. If standardized within ISO, OASIS will handle new item requests as appropriate.	Accepted	
JP-06	Explanatory Report		ge	4) New version of UOML When OASIS publishes a new version of UOML, it should be submitted to JTC1 using the PAS procedure again. If this is the case, member bodies may submit comments as part of their DIS votes.	None. **OASIS UOML-X TC Response** **Accepted**. If standardized	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				How will such comments be addressed? Will the OASIS standard be changed or further errata be created? Follow the best practises being established by JTC1, SC34, and OASIS.	within ISO, OASIS will follow proper procedure as appropriate.		
JP-07	Explanatory Report		te	5) References to schemas and examples Schemas and examples referenced from the explanatory report should be included as part of the DIS.	None. **OASIS UOML-X TC Response** **Accepted**. The UOML specification does provide a normative XML Schema Definition (XSD). A new more detailed example clause has been added to the specification, however.	Partially Accepted with Change	See JP-10
JP-08	BNF notation		te	The BNF notation should be described in accordance with ISO/IEC 14977.	None. **OASIS UOML-X TC Response.** **Accepted**. Added a normative reference to ISO/IEC 14977 in the Normative Reference clause. See JP-08 details for more information.	Accepted with Change	Proposal to modify the current BNF to use the current ISO standard for BNF and ensure there are no errors.
JP-09	API description		te	The API should be described in accordance with CORBA IDL (ISO/IEC 10728/Amd.3).	None. **OASIS UOML-X TC Response**	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Rejected. Both XML and IDL can be used to describe the API; however, XML provides a better compatibility between different versions of the standard. For example, if an additional parameter needs to be added for an instruction in the UOML standard, in the XML case, an application which understands the new standard can call the DCMS which does not understand the new standard without issue; while in the IDL case, a new interface should be added and if an application calls a DCMS which does not understand the new standard, a run-time error will be generated. The structure of the normative XML schema is defined in accordance with the XML schema standard defined by W3C, a valid standard to be normatively referenced.		
JP-10	Structure of XML		te	The structures of XML should be described in accordance with ISO/IEC 19757.	None. **OASIS UOML-X TC Response**	Partially Accepted with Change	Proposal to add a RELAX NG representation of the UOML XML schema specification as an informative annex (using

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					Rejected. The structure of the normative XML schema is defined in accordance with the XML and XML schema standards defined by W3C, both valid standards to be normatively referenced.		compact syntax).
KR-01	3.1		TE	There is not provide the explanation of the exact mark scheme(URL, URI etc.) about path attribute-value of OPEN elements. It gives the definition of Return value, but it isn't also exist the definition of HANDLE which explain the return value. Also, No logic and scheme is presented as RET and it may be difficult to use for real user of this standards due to lack of information.	None. **OASIS UOML-X TC Response** **Accepted**. Provided further explanation regarding the path property of OPEN. HANDLE and RET are defined in the normative UOML XML schema provided with the specification and also now added as an informative Appendix to the UOML specification. RET is also defined as a separate instruction within the UOML specification as well. See KR01 details for more information.	Rejected	
KR-02	3.2		TE	It gives the explanation of handle as a character string value, representing the handle of Docbase to be closed, but it may cause the problem of use for various purpose by user because of no exist the scheme of string value.	None. **OASIS UOML-X TC Response**	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				It could access documents and contents in the local using the command such as OPEN, CLOSE. However, there is not method to mark the attribute or option requested by various environments relating security.	**Rejected**. The handle is defined as xs:string via the XML Schema specification. Security is outside the scope of this Part of UOML, but may be covered in a future Part.		
KR-03	3.3		GE	There is no exist the exactly definition of current object	None. **OASIS UOML-X TC Response** **Accepted**. Added an informative note describing object and current object. See KR-03 details for more information.	Accepted	
KR-04	3.4		TE	It gives the definition of operation method using the GET_SUB, GET_SUB_COUNT, GET_PROP, GET_PAGE_BMP value, but there is not explained the detailed sub-elements and is not clear the unit of attribute-value used in the sub elements.	None. **OASIS UOML-X TC Response** **Accepted**. All units are defined in the normative XML Schema accompanying the UOML specification. Unit information has also been added to the GET_SUB, GET_SUB_COUNT, GET_PROP, GET_PAGE_BMP within the	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					revised specification. See KR-04 details for more information.		
NL-01	1.2 Overview		GE	The scope of the standard is not clear.	Please make the scope more concrete. **OASIS UOML-X TC Response** **Accepted**. Added a Scope clause to the specification. See NL01 details for more information.	Accepted	
NL-02	Whole document		TE	To many undefined terms and examples rather than normative references.	None **OASIS UOML-X TC Response** **Accepted**. Normative Reference clause and Terminology Definitions clause were in the original specification. New definitions of terms and examples will be added to the specification.	Accepted	
NL-03	Whole document		TE	Because of the lack of detail the standard can't be implemented.	None. **OASIS UOML-X TC Response** **Accepted**. Expanded the Conformance clause, provided more information and examples in order to aid	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					in possible implementations. See NL-03 details for more information.		
NL-04	Whole document		GE	In general, the quality of the standard is not good.	None. **OASIS UOML-X TC Response** **Accepted**. Restructured and reformatted the specification to provide better quality. The specification has been revised editorially and technical details have been strengthened.	Accepted	
NO-01	All		ge	We do not feel that this specification is of the quality required for it to become an International Standard. Further, we feel that it is not sufficiently clear what problem the specification exists to solve. Therefore, if SC34 is to perform further work on this specification, the first step should be to clarify the background and requirements for this proposed standard.	Clarify purpose and justification for a standard. **OASIS UOML-X TC Response** **Accepted**. Added new clauses to the specification (e.g., Scope) and modified other clauses for better clarity. See NO-01 details for more information.	Rejected	
NO-02	All		ge	Lack of Scope statement: Due to lack of	Produce a clearly stated	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				scope statement the purpose of this standard is unclear.	scope statement. **OASIS UOML-X TC Response** **Accepted**. Added a Scope clause to the specification. See NO-02 details for more information.		
NO-03	All		ge	Format and structure of document.	Format the document in conformance to ISO/IEC Directives. **OASIS UOML-X TC Response** **Accepted**. Document has been modified to conform better to the ISO directives. However, since this is an initial OASIS submission, the primary structure still remains OASIS'.	Accepted	
NO-04	1.1	Keywords:	te	Keywords: is not the proper phrase to use – what is mentioned below is not keywords.	Remove text: "Keywords:" **OASIS UOML-X TC Response** **Accepted**. Replaced "keywords" with "terminology". However, "keywords" is a proper word to use and this change isn't necessarily mandatory because these words come from the RFC 'Keywords for	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					use in RFCs to indicate Requirement Levels"		
NO-05	1.1	Docset:	te	The explanation of "Docset" does not convey any meaning. In our interpretation a "directory in filesystem" is describing a hierarchical structure, containing documents. If a docset is is a set of documents distributed in a hierarchical structure this it should be described so.	Rewrite definition of "docset" **OASIS UOML-X TC Response** **Accepted**. Clarified the definition of both docbase and docset within the Terminology clause. See NO-05 details for more information.	Rejected	
NO-06	1.1	Path	te	This definition does not convey any meaning. As written now a Path could be anything. It is also unclear what a "region collection" is.	Rewrite definition of "Path". **OASIS UOML-X TC Response** **Accepted**. Clarified the definition of Path within the Terms and Definition clause. See NO-06 details for more information.	Rejected	
NO-07	1.1		te	Several terms used within the definitions of the terminology in this document is not well defined, and should have a definition to be used.	Add definition for: – "Region collection" – Operation – Document – "Line/Curve segment" – "Upper level	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					object” – “application software” – “process documents” **OASIS UOML-X TC Response** **Accepted**. Added a definition for “application software” within the Terminology clause. All other terms are self-explanatory within the context of their usage. “upper level object” has been stricken from the specification. See NO-07 details for more information.		
NO-08	1.1	Command Object	te	It is unclear if a Command object is used for modifying or “holding” graphics control parameters.	Rewrite definition **OASIS UOML-X TC Response** **Accepted**. The definition of command object was rewritten for better clarity within the Terminology clause. See NO-08 details for more information.	Rejected	
NO-09	1.2	1st Para	te	It is unclear what the meaning of this 1st paragraph is. And by reading the rest of the	Remove paragraph	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				document, we have concerns on the validity of the statements put forward in the paragraph.	**OASIS UOML-X TC Response** **Accepted**. Rewrote the wording of this paragraph. See NO-09 details for more information.		
NO-10	1.2	2nd Para	te	Due to lack of Scope of this standard, we have problems understanding what is meant by this paragraph. Is this a standard for representing pixels in a “two dimensions” on a page. And if this standard is about “Static paging information” we have problems understanding what type of “UOML defined functions” an application can use to process this static document?	Remove paragraph, or rewrite. **OASIS UOML-X TC Response** **Accepted**. Added a scope clause to this standard and also modified the paragraph in question for better clarity. Also the Overview clause where this paragraph is found is informative only. See NO-10 details for more information.	Accepted	
NO-11	1.2	3rd Para	te	It is unclear what is meant by this paragraph. The paragraph should be restructured to better explain what is meant.	Remove paragraph, or rewrite. **OASIS UOML-X TC Response** **Accepted**. The paragraph as it was has been removed and replaced with multiple paragraphs for better clarity.	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					See NO-11 details for more information.		
NO-12	1.2	5th Para	ed	Sentence as introduction to rest of document is unnecessary	Remove paragraph. **OASIS UOML-X TC Response** **Accepted**. Removed the final sentence of the Overview.	Accepted	
NO-13	1.3	All	te	Normative references should be normative references i.e. a normative reference is a standard that is a necessary standard for implementing this one.	Only include standards that are regarded as necessary for implementation of this standard. **OASIS UOML-X TC Response** **Rejected**. It is believed that all current normative references are necessary for the implementation of this standard.	Accepted	
NO-14	1.3	[PNG, JBIG, JPEG]	te	By looking at the document, we have problems understanding why PNG, JBIG, JPEG is normative references, and how this standard is based on these standards?	Remove the normative references. **OASIS UOML-X TC Response** **Rejected**. For the IMAGE element, PNG, JPEG, etc. are possible image type values.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
NO-15	2	1st Para	te	The sentence “The abstract document model can be regarded as…” Either a document model is or it is not, it would be difficult if it could be regarded as something.	Rewrite sentence to make it clear what an “abstract document model” is. **OASIS UOML-X TC Response** **Accepted**. Reworded the sentence to say “is” instead of “can be regarded as”	Rejected	
NO-16	2	2nd Para	te	A standard do not have issues.	Rewrite sentence. **OASIS UOML-X TC Response** **Accepted**. Used “areas” instead of “issues”	Rejected	
NO-17	2, 3	All	te	Since the standard is lacking several important definitions, and since the Scope of the standard is not specified it is hard to make detailed comments on these parts. Based on the reading of the document, we find that there is a mix prosaic descriptions and illustrations which makes this document hard to interpret in a consistent way.	We would recommend that this project is cancelled, and that proper use-cases and rationale for this work is brought forward as a NWI proposal. Such a NWI proposal should provide a clear scope and rationale and the business need for this standard. **OASIS UOML-X TC Response** **Accepted**. Added a scope, more illustrative examples, ensured the UOML XML	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
					schema (XSD) is part of the normative standard, etc. in order to make this a more clear standard. Provided a business case of UOML as part of this disposition of comments. See NO-17 details for more information.		
NO-18	3		ge	UOML appears to be a protocol or web service interface for introspecting and modifying documents, but the operations are defined as XML elements, with no specified mechanism for passing these elements to a UOML implementation. Thus, while UOML superficially may appear to specify an interface, it in reality does no such thing, making it impossible to actually create interoperable implementations. This flaw is so serious as to make the specification as it stands unacceptable.	None. **OASIS UOML-X TC Response** **Rejected**. The interface is normatively specified as an XSD via the accepted XML Schema standard. This Part only defines the instructions. The passing of instructions will be defined in future Parts of the UOML specification.	Accepted with Change	Proposal to add to the Scope clause “This standard does not define any binding for the operations on the UOML document model. Such bindings are implementation-defined or will be defined in other parts of this standard”.
NO-19	3		te	The XML elements that represent operations are not defined using an XML schema language, but instead in prose. This makes the specificiation unnecessarily vague and hard to read. Further, the prose is not consistent for all operations. This must change if the specification is to be made acceptable.	None. **OASIS UOML-X TC Response** **Accepted**. The XML Schema Definition (XSD) is a normative part of the UOML specification.	Accepted	
NO-20	3		te	The UOML document model is entirely at odds with those of XML documents in general, and of OOXML and ODF	None. **OASIS UOML-X TC**	Rejected	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

1	2	(3)	5	6	7	8	9
Code of MB[1]	**Clause No./ Subclause No./ Annex**	**Paragraph/ Figure/Table/Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the MB**	**Proposed change by the MB**	**SC34 NB Response To OASIS Response**	**SC34 NB Additional Proposal(s)**
				documents, since it is page-based. This means that it is not even remotely clear how UOML can be used to process OOXML and ODF documents, making its relevance to SC34 uncertain. It might perhaps be used for processing PDF documents, but if so the specification should move to the SC responsible for PDF.	**Response** **Rejected**. UOML is an interface standard to process documents, like PDF. However, PDF is not standardized within JTC1. SC34 is the TC within JTC1 that is most suitable for UOML standardization.		
NO-21	3		te	The operations are not very cleanly defined. For example, to delete a document one uses an OPEN operation with the second parameter set to "false". Why not use the DELETE operation for this? Further, the definition of many of these operations is far from clear. What does the OPEN operation do? This is not clearly specified, and the whole operation gives the impression of being tied to a specific implementation. These are serious flaws which must be remedied before the specification can be accepted.	None. **OASIS UOML-X TC Response** **Rejected**. An OPEN operation with the second parameter set to “false” is not to delete a docbase. Rather it is used to specify whether to create a new docbase, even if the docbase already exists. However, clarifying statements and examples have been added as needed.	Accepted	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

(back to table)

GENERAL COMMENT:
There were many comments by the NBs about lack of data type specification, etc. The original submission mistakenly did not point to the UOML

The original submission mistakenly did not point to the UOML XML Schema definition. Also, the XSD was mistakenly omitted in the original submission as well. The UOML XML Schema definition has been added to the Normative References sub-clause.

[UOMLSchema] *UOML Part 1 v1.0 Schema*, http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd

In various, relevant clauses, including the introduction to the UOML Instructions and UOML Objects clauses, a normative statement has been added. For example:

The formal definitions for these instructions are specified in the UOML XML Schema Definition located at: http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd.

An informative Annex was added to the specification copying the contents of the .xsd file for easy spec reference. The .xsd file, however, is the normative specification of the UOML schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:uoml="urn:oasis:names:tc:uoml:xmlns:uoml:1.0"
targetNamespace="urn:oasis:names:tc:uoml:xmlns:uoml:1.0" elementFormDefault="unqualified" attributeFormDefault="unqualified">
	<xs:complexType name="ARC">
		<xs:annotation>
			<xs:documentation>arc</xs:documentation>
		</xs:annotation>
		<xs:attribute name="clockwise" type="xs:boolean" use="required"/>
		<xs:attribute name="start" type="xs:string" use="required"/>
		<xs:attribute name="end" type="xs:string" use="required"/>
		<xs:attribute name="center" type="xs:string" use="required"/>
		<xs:attribute name="angle" type="xs:float" use="required"/>
	</xs:complexType>
	<xs:complexType name="BEZIER">
		<xs:annotation>
			<xs:documentation>bezier curve</xs:documentation>
		</xs:annotation>
		<xs:attribute name="start" type="xs:string" use="required"/>
```

```
39              <xs:attribute name="ctrl" type="xs:string" use="required"/>
40              <xs:attribute name="ctrl2" type="xs:string" use="optional"/>
41              <xs:attribute name="end" type="xs:string" use="required"/>
42       </xs:complexType>
43       <xs:complexType name="CIRCLE">
44              <xs:annotation>
45                     <xs:documentation>circle</xs:documentation>
46              </xs:annotation>
47              <xs:attribute name="radius" type="xs:int" use="required"/>
48              <xs:attribute name="center" type="xs:string" use="required"/>
49       </xs:complexType>
50       <xs:complexType name="LINE">
51              <xs:annotation>
52                     <xs:documentation>line</xs:documentation>
53              </xs:annotation>
54              <xs:attribute name="start" type="xs:string" use="required"/>
55              <xs:attribute name="end" type="xs:string" use="required"/>
56       </xs:complexType>
57       <xs:complexType name="RECT">
58              <xs:annotation>
59                     <xs:documentation>rect</xs:documentation>
60              </xs:annotation>
61              <xs:attribute name="tl" type="xs:string" use="required"/>
62              <xs:attribute name="br" type="xs:string" use="required"/>
63       </xs:complexType>
64       <xs:complexType name="ELLIPSE">
65              <xs:annotation>
66                     <xs:documentation>ellipse</xs:documentation>
67              </xs:annotation>
68              <xs:attribute name="xr" type="xs:int" use="required"/>
69              <xs:attribute name="yr" type="xs:int" use="required"/>
70              <xs:attribute name="center" type="xs:string" use="required"/>
71              <xs:attribute name="angle" type="xs:float" use="required"/>
72       </xs:complexType>
73       <xs:complexType name="ROUNDRECT">
74              <xs:annotation>
75                     <xs:documentation>roundrect</xs:documentation>
76              </xs:annotation>
```

```
		<xs:attribute name="xr" type="xs:int" use="required"/>
		<xs:attribute name="yr" type="xs:int" use="required"/>
		<xs:attribute name="tl" type="xs:string" use="required"/>
		<xs:attribute name="br" type="xs:string" use="required"/>
	</xs:complexType>
	<xs:complexType name="META">
		<xs:annotation>
			<xs:documentation>metadata</xs:documentation>
		</xs:annotation>
		<xs:attribute name="key" type="xs:string" use="required"/>
		<xs:attribute name="val" type="xs:string" use="required"/>
	</xs:complexType>
	<xs:complexType name="METALIST">
		<xs:annotation>
			<xs:documentation>metadata list</xs:documentation>
		</xs:annotation>
		<xs:sequence>
			<xs:element name="meta" type="uoml:META" minOccurs="0" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<xs:complexType name="CMD">
		<xs:annotation>
			<xs:documentation>cmd</xs:documentation>
		</xs:annotation>
		<xs:sequence minOccurs="0">
			<xs:choice>
				<xs:element name="cliparea" type="uoml:PATH"/>
				<xs:element name="matrix" type="uoml:MATRIX"/>
				<xs:element name="rgb" type="uoml:COLOR_RGB"/>
			</xs:choice>
		</xs:sequence>
		<xs:attribute name="name" type="uoml:CMDNAME" use="required"/>
		<xs:attribute name="v1" type="xs:anySimpleType"/>
		<xs:attribute name="v2" type="xs:anySimpleType"/>
	</xs:complexType>
	<xs:complexType name="MATRIX">
		<xs:annotation>
			<xs:documentation>matrix</xs:documentation>
```

```
		</xs:annotation>
		<xs:attribute name="f11" type="xs:float" use="required"/>
		<xs:attribute name="f12" type="xs:float" use="required"/>
		<xs:attribute name="f21" type="xs:float" use="required"/>
		<xs:attribute name="f22" type="xs:float" use="required"/>
		<xs:attribute name="f31" type="xs:float" use="required"/>
		<xs:attribute name="f32" type="xs:float" use="required"/>
	</xs:complexType>
	<xs:complexType name="SUBPATH">
		<xs:annotation>
			<xs:documentation>subpath</xs:documentation>
		</xs:annotation>
		<xs:attribute name="data" type="xs:string" use="required"/>
	</xs:complexType>
	<xs:complexType name="PATH">
		<xs:annotation>
			<xs:documentation>path</xs:documentation>
		</xs:annotation>
		<xs:sequence>
			<xs:choice minOccurs="0" maxOccurs="unbounded">
				<xs:element name="subpath" type="uoml:SUBPATH"/>
				<xs:element name="rect" type="uoml:RECT"/>
				<xs:element name="circle" type="uoml:CIRCLE"/>
				<xs:element name="ellipse" type="uoml:ELLIPSE"/>
				<xs:element name="roundrect" type="uoml:ROUNDRECT"/>
			</xs:choice>
		</xs:sequence>
	</xs:complexType>
	<xs:complexType name="COLOR_RGB">
		<xs:annotation>
			<xs:documentation>rgb color</xs:documentation>
		</xs:annotation>
		<xs:attribute name="r" type="xs:short" use="required"/>
		<xs:attribute name="g" type="xs:short" use="required"/>
		<xs:attribute name="b" type="xs:short" use="required"/>
		<xs:attribute name="a" type="xs:short" use="optional"/>
	</xs:complexType>
	<xs:complexType name="EMBEDFONT">
```

```
153             <xs:annotation>
154                 <xs:documentation>embedded font</xs:documentation>
155             </xs:annotation>
156             <xs:simpleContent>
157                 <xs:extension base="xsd:base64Binary">

158                 </xs:extension>
159             </xs:simpleContent>
160         </xs:complexType>
161         <xs:complexType name="FONTMAP">
162             <xs:annotation>
163                 <xs:documentation>font mapping</xs:documentation>
164             </xs:annotation>
165             <xs:attribute name="name" type="xs:string" use="required"/>
166             <xs:attribute name="no" type="xs:int" use="required"/>
167         </xs:complexType>
168         <xs:complexType name="FONTLIST">
169             <xs:annotation>
170                 <xs:documentation>font list</xs:documentation>
171             </xs:annotation>
172         </xs:complexType>
173         <xs:complexType name="IMAGE">
174             <xs:annotation>
175                 <xs:documentation>image</xs:documentation>
176             </xs:annotation>
177             <xs:simpleContent>
178                 <xs:extension base="xsd:base64Binary">
179                     <xs:attribute name="tl" type="xs:string" use="required"/>
180                     <xs:attribute name="br" type="xs:string" use="required"/>
181                     <xs:attribute name="type" type="xs:string" use="required"/>
182                     <xs:attribute name="path" type="xs:string" use="optional"/>
183                 </xs:extension>
184             </xs:simpleContent>

185     </xs:complexType>
186         <xs:complexType name="TEXT">
187             <xs:annotation>
188                 <xs:documentation>text</xs:documentation>
189             </xs:annotation>
190             <xs:attribute name="origin" type="xs:string" use="required"/>
```

```
191             <xs:attribute name="encode" type="xs:string" use="required"/>
192             <xs:attribute name="text" type="xs:string" use="required"/>
193             <xs:attribute name="spaces" type="xs:string" use="optional"/>
194         </xs:complexType>
195         <xs:simpleType name="CMDNAME">
196             <xs:annotation>
197                 <xs:documentation>command names</xs:documentation>
198             </xs:annotation>
199             <xs:restriction base="xs:string">
200                 <xs:enumeration value="COLOR_LINE"/>
201                 <xs:enumeration value="COLOR_FILL"/>
202                 <xs:enumeration value="COLOR_TEXT"/>
203                 <xs:enumeration value="COLOR_SHADOW"/>
204                 <xs:enumeration value="COLOR_OUTLINE"/>
205                 <xs:enumeration value="LINE_WIDTH"/>
206                 <xs:enumeration value="LINE_JOIN"/>
207                 <xs:enumeration value="LINE_CAP"/>
208                 <xs:enumeration value="MITER_LIMIT"/>
209                 <xs:enumeration value="FILL_RULE"/>
210                 <xs:enumeration value="RENDER_MODE"/>
211                 <xs:enumeration value="RASTER_OP"/>
212                 <xs:enumeration value="TEXT_DIR"/>
213                 <xs:enumeration value="CHAR_DIR"/>
214                 <xs:enumeration value="CHAR_ROTATE"/>
215                 <xs:enumeration value="CHAR_SLANT"/>
216                 <xs:enumeration value="CHAR_SIZE"/>
217                 <xs:enumeration value="CHAR_WEIGHT"/>
218                 <xs:enumeration value="CHAR_STYLE"/>
219                 <xs:enumeration value="TEXT_MATRIX"/>
220                 <xs:enumeration value="IMAGE_MATRIX"/>
221                 <xs:enumeration value="GRAPH_MATRIX"/>
222                 <xs:enumeration value="EXT_MATRIX"/>
223                 <xs:enumeration value="PUSH_GS"/>
224                 <xs:enumeration value="POP_GS"/>
225                 <xs:enumeration value="SHADOW_WIDTH"/>
226                 <xs:enumeration value="SHADOW_DIR"/>
227                 <xs:enumeration value="SHADOW_LEN"/>
228                 <xs:enumeration value="SHADOW_NEG"/>
229                 <xs:enumeration value="SHADOW_ATL"/>
230                 <xs:enumeration value="CLIP_AREA"/>
231                 <xs:enumeration value="FONT"/>
232                 <xs:enumeration value="OUTLINE_BORDER"/>
233                 <xs:enumeration value="OUTLINE_WIDTH"/>
234                 <xs:enumeration value="HOLLOW_BORDER"/>
```

```
235             </xs:restriction>
236         </xs:simpleType>
237         <xs:simpleType name="LINECAP">
238             <xs:annotation>
239                 <xs:documentation>line cap style</xs:documentation>
240             </xs:annotation>
241             <xs:restriction base="xs:string">
242                 <xs:enumeration value="END_BUTT"/>
243                 <xs:enumeration value="END_SQUARE"/>
244                 <xs:enumeration value="END_ROUND"/>
245             </xs:restriction>
246         </xs:simpleType>
247         <xs:simpleType name="JOINCAP">
248             <xs:annotation>
249                 <xs:documentation>line join style</xs:documentation>
250             </xs:annotation>
251             <xs:restriction base="xs:string">
252                 <xs:enumeration value="JOIN_MITER"/>
253                 <xs:enumeration value="JOIN_BEVEL"/>
254                 <xs:enumeration value="JOIN_ROUND"/>
255             </xs:restriction>
256         </xs:simpleType>
257         <xs:simpleType name="FILLRULE">
258             <xs:annotation>
259                 <xs:documentation>fill rule</xs:documentation>
260             </xs:annotation>
261             <xs:restriction base="xs:string">
262                 <xs:enumeration value="RULE_EVENODD"/>
263                 <xs:enumeration value="RULE_WINDING"/>
264             </xs:restriction>
265         </xs:simpleType>
266         <xs:simpleType name="ROP">
267             <xs:annotation>
268                 <xs:documentation>rop operation</xs:documentation>
269             </xs:annotation>
270             <xs:restriction base="xs:string">
271                 <xs:enumeration value="ROP_COPY"/>
272                 <xs:enumeration value="ROP_N_COPY"/>
273                 <xs:enumeration value="ROP_RESET"/>
274                 <xs:enumeration value="ROP_SET"/>
275                 <xs:enumeration value="ROP_NOP"/>
276                 <xs:enumeration value="ROP_REV"/>
277                 <xs:enumeration value="ROP_AND"/>
278                 <xs:enumeration value="ROP_AND_N"/>
```

```
                <xs:enumeration value="ROP_N_AND"/>
                <xs:enumeration value="ROP_N_AND_N"/>
                <xs:enumeration value="ROP_OR"/>
                <xs:enumeration value="ROP_OR_N"/>
                <xs:enumeration value="ROP_N_OR"/>
                <xs:enumeration value="ROP_N_OR_N"/>
                <xs:enumeration value="ROP_XOR"/>
                <xs:enumeration value="ROP_EOR"/>
        </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CHARTXTDIR">
        <xs:annotation>
                <xs:documentation>text or char direction</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
                <xs:enumeration value="HEAD_LEFT"/>
                <xs:enumeration value="HEAD_RIGHT"/>
                <xs:enumeration value="HEAD_TOP"/>
                <xs:enumeration value="HEAD_BOTTOM"/>
        </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SHADOWDIR">
        <xs:annotation>
                <xs:documentation>shadow direction</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
                <xs:enumeration value="SHADOW_LT"/>
                <xs:enumeration value="SHADOW_LB"/>
                <xs:enumeration value="SHADOW_RT"/>
                <xs:enumeration value="SHADOW_RB"/>
        </xs:restriction>
</xs:simpleType>
<xs:complexType name="OBJSTREAM">
        <xs:annotation>
                <xs:documentation>object stream</xs:documentation>
        </xs:annotation>
</xs:complexType>
<xs:complexType name="LAYER">
        <xs:annotation>
                <xs:documentation>layer</xs:documentation>
        </xs:annotation>
</xs:complexType>
<xs:complexType name="PAGE">
        <xs:annotation>
```

```
                <xs:documentation>page</xs:documentation>
        </xs:annotation>
        <xs:attribute name="width" type="xs:float" use="required"/>
        <xs:attribute name="height" type="xs:float" use="required"/>
        <xs:attribute name="resolution" type="xs:int" use="required"/>
</xs:complexType>
<xs:complexType name="DOC">
        <xs:annotation>
                <xs:documentation>doc</xs:documentation>
        </xs:annotation>
        <xs:sequence>
                <xs:element name="metainfo" type="uoml:METALIST"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="DOCSET">
        <xs:annotation>
                <xs:documentation>doc set</xs:documentation>
        </xs:annotation>
        <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="DOCBASE">
        <xs:annotation>
                <xs:documentation>doc base</xs:documentation>
        </xs:annotation>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="path" type="xs:string" use="required"/>
</xs:complexType>
<xs:element name="CLOSE">
        <xs:complexType>
                <xs:attribute name="handle" type="xs:string" use="optional"/>
        </xs:complexType>
</xs:element>
<xs:element name="DELETE">
        <xs:complexType>
                <xs:attribute name="handle" type="xs:string" use="optional"/>
        </xs:complexType>
</xs:element>
<xs:element name="INSERT">
        <xs:complexType>
                <xs:choice>
                        <xs:element name="xobj" type="uoml:COMPOUND"/>
                </xs:choice>
                <xs:attribute name="handle" type="xs:string"/>
```

```
                <xs:attribute name="pos" type="xs:int"/>
        </xs:complexType>
</xs:element>
<xs:element name="GET">
        <xs:complexType>
                <xs:choice>
                        <xs:element name="disp_conf">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="clip" type="uoml:PATH" minOccurs="0"/>
                                        </xs:sequence>
                                        <xs:attribute name="end_layer" type="xs:int"/>
                                        <xs:attribute name="resolution" type="xs:int"/>
                                        <xs:attribute name="format" type="xs:string"/>
                                        <xs:attribute name="output" type="xs:string" use="required"/>
                                        <xs:attribute name="addr" type="xs:string" use="required"/>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="pos">
                                <xs:complexType>
                                        <xs:attribute name="val" type="xs:int" use="required"/>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="property">
                                <xs:complexType>
                                        <xs:attribute name="name" type="xs:string" use="required"/>
                                </xs:complexType>
                        </xs:element>
                </xs:choice>
                <xs:attribute name="usage" type="xs:string" use="required"/>
                <xs:attribute name="handle" type="xs:string"/>
        </xs:complexType>
</xs:element>
<xs:element name="SET">
        <xs:complexType>
                <xs:choice>
                        <xs:choice minOccurs="0" maxOccurs="unbounded">
                                <xs:element name="intVal" type="uoml:INT"/>
                                <xs:element name="floatVal" type="uoml:DOUBLE"/>
                                <xs:element name="timeVal" type="uoml:TIME"/>
                                <xs:element name="dateVal" type="uoml:DATE"/>
                                <xs:element name="dateTimeVal" type="uoml:DATETIME"/>
                                <xs:element name="durationVal" type="uoml:DURATION"/>
                                <xs:element name="stringVal" type="uoml:STRING"/>
```

```
                                <xs:element name="binaryVal" type="uoml:BINARY"/>
                                <xs:element name="compoundVal" type="uoml:COMPOUND"/>
                                <xs:element name="boolVal" type="uoml:BOOL"/>
                        </xs:choice>
                </xs:choice>
                <xs:attribute name="handle" type="xs:string"/>
        </xs:complexType>
</xs:element>
<xs:element name="USE">
        <xs:complexType>
                <xs:attribute name="handle" type="xs:string" use="required"/>
        </xs:complexType>
</xs:element>
<xs:element name="OPEN">
        <xs:complexType>
                <xs:attribute name="create" type="xs:boolean" default="true"/>
                <xs:attribute name="del_exist" type="xs:boolean" default="false"/>
                <xs:attribute name="path" type="xs:string" use="required"/>
        </xs:complexType>
</xs:element>
<xs:element name="SYSTEM">
        <xs:complexType>
                <xs:choice>
                        <xs:element name="flush">
                                <xs:complexType>
                                        <xs:attribute name="handle"/>
                                        <xs:attribute name="path"/>
                                </xs:complexType>
                        </xs:element>
                </xs:choice>
        </xs:complexType>
</xs:element>
<xs:element name="RET">
        <xs:complexType>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="intVal" type="uoml:INT"/>
                        <xs:element name="floatVal" type="uoml:DOUBLE"/>
                        <xs:element name="timeVal" type="uoml:TIME"/>
                        <xs:element name="dateVal" type="uoml:DATE"/>
                        <xs:element name="dateTimeVal" type="uoml:DATETIME"/>
                        <xs:element name="durationVal" type="uoml:DURATION"/>
                        <xs:element name="stringVal" type="uoml:STRING"/>
                        <xs:element name="binaryVal" type="uoml:BINARY"/>
                        <xs:element name="compoundVal" type="uoml:COMPOUND"/>
```

```
                        <xs:element name="boolVal" type="uoml:BOOL"/>
                        <xs:element name="longVal" type="uoml:LONG"/>
                </xs:choice>
        </xs:complexType>
</xs:element>
<xs:complexType name="COMPOUND">
        <xs:annotation>
                <xs:documentation>compound parameter type</xs:documentation>
        </xs:annotation>
        <xs:choice minOccurs="0">
                <xs:element name="arc" type="uoml:ARC"/>
                <xs:element name="bezier" type="uoml:BEZIER"/>
                <xs:element name="circle" type="uoml:CIRCLE"/>
                <xs:element name="cmd" type="uoml:CMD"/>
                <xs:element name="rgb" type="uoml:COLOR_RGB"/>
                <xs:element name="doc" type="uoml:DOC"/>
                <xs:element name="docbase" type="uoml:DOCBASE"/>
                <xs:element name="docset" type="uoml:DOCSET"/>
                <xs:element name="ellipse" type="uoml:ELLIPSE"/>
                <xs:element name="embedfont" type="uoml:EMBEDFONT"/>
                <xs:element name="fontlist" type="uoml:FONTLIST"/>
                <xs:element name="fontmap" type="uoml:FONTMAP"/>
                <xs:element name="image" type="uoml:IMAGE"/>
                <xs:element name="layer" type="uoml:LAYER"/>
                <xs:element name="line" type="uoml:LINE"/>
                <xs:element name="matrix" type="uoml:MATRIX"/>
                <xs:element name="meta" type="uoml:META"/>
                <xs:element name="metalist" type="uoml:METALIST"/>
                <xs:element name="page" type="uoml:PAGE"/>
                <xs:element name="path" type="uoml:PATH"/>
                <xs:element name="rect" type="uoml:RECT"/>
                <xs:element name="roundrect" type="uoml:ROUNDRECT"/>
                <xs:element name="subpath" type="uoml:SUBPATH"/>
                <xs:element name="text" type="uoml:TEXT"/>
                <xs:element name="objstream" type="uoml:OBJSTREAM"/>
        </xs:choice>
        <xs:attribute name="name" type="xs:string"/>
</xs:complexType>
<xs:complexType name="STRING">
        <xs:annotation>
                <xs:documentation>string parameter type</xs:documentation>
        </xs:annotation>
        <xs:attribute name="val" type="xs:string" use="required"/>
        <xs:attribute name="name" type="xs:string"/>
```

```
499         </xs:complexType>
500         <xs:complexType name="DOUBLE">
501                 <xs:annotation>
502                         <xs:documentation>double precision float parameter type</xs:documentation>
503                 </xs:annotation>
504                 <xs:attribute name="val" type="xs:double" use="required"/>
505                 <xs:attribute name="name" type="xs:string"/>
506         </xs:complexType>
507         <xs:complexType name="DATE">
508                 <xs:annotation>
509                         <xs:documentation>date parameter type</xs:documentation>
510                 </xs:annotation>
511                 <xs:attribute name="val" type="xs:date" use="required"/>
512                 <xs:attribute name="name" type="xs:string"/>
513         </xs:complexType>
514         <xs:complexType name="DATETIME">
515                 <xs:annotation>
516                         <xs:documentation>date and time parameter type</xs:documentation>
517                 </xs:annotation>
518                 <xs:attribute name="val" type="xs:dateTime" use="required"/>
519                 <xs:attribute name="name" type="xs:string"/>
520         </xs:complexType>
521         <xs:complexType name="TIME">
522                 <xs:annotation>
523                         <xs:documentation>time parameter type</xs:documentation>
524                 </xs:annotation>
525                 <xs:attribute name="val" type="xs:time" use="required"/>
526                 <xs:attribute name="name" type="xs:string"/>
527         </xs:complexType>
528         <xs:complexType name="DURATION">
529                 <xs:annotation>
530                         <xs:documentation>duration parameter type</xs:documentation>
531                 </xs:annotation>
532                 <xs:attribute name="val" type="xs:duration" use="required"/>
533                 <xs:attribute name="name" type="xs:string"/>
534         </xs:complexType>
535         <xs:complexType name="BINARY">
536                 <xs:annotation>
537                         <xs:documentation>binary parameter type</xs:documentation>
538                 </xs:annotation>
539                 <xs:attribute name="val" type="xs:base64Binary" use="required"/>
540                 <xs:attribute name="name" type="xs:string"/>
541         </xs:complexType>
542         <xs:complexType name="INT">
```

```
543                 <xs:annotation>
544                         <xs:documentation>integer parameter type</xs:documentation>
545                 </xs:annotation>
546                 <xs:attribute name="val" type="xs:int" use="required"/>
547                 <xs:attribute name="name" type="xs:string"/>
548         </xs:complexType>
549         <xs:complexType name="BOOL">
550                 <xs:annotation>
551                         <xs:documentation>boolean parameter type</xs:documentation>
552                 </xs:annotation>
553                 <xs:attribute name="val" type="xs:boolean" use="required"/>
554                 <xs:attribute name="name" type="xs:string"/>
555         </xs:complexType>
556         <xs:complexType name="LONG">
557                 <xs:annotation>
558                         <xs:documentation>long parameter type</xs:documentation>
559                 </xs:annotation>
560                 <xs:attribute name="name" type="xs:string"/>
561                 <xs:attribute name="val" type="xs:long" use="required"/>
562         </xs:complexType>
563         <xs:simpleType name="CHARSTYLE">
564                 <xs:restriction base="xs:string">
565                         <xs:enumeration value="SHADOW"/>
566                         <xs:enumeration value="HOLLOW"/>
567                         <xs:enumeration value="OUTLINE"/>
568                 </xs:restriction>
569         </xs:simpleType>
570 </xs:schema>
```

(back to table)

CZ-02:
There are many typos in text of the submission. In many cases the text does not follow grammar rules of the English language and it is impossible to interpret it in an unambiguous way which is a must for an international standard. Page 8 section 1.2 the third edition of XML specification is referenced although the fourth edition is the one which is endorsed by SC34 for use in international standards produced within SC34. Page 8 section 1.2 there is a reference [xmlschema-1] pointing to W3C XML Schema version 1.1 – this version of XML schema is still in draft stages at W3C and should not be referenced from other specifications including the submission.

The specification has been improved grammatically. Made changes to the normative references to XML. The XML Schema reference is pointing to the correct XML Schema specification already.

XML, Tim Bray, Eve Maler, Jean Paoli, C. M. Sperberg-McQueen, François Yergeau (editors). *Extensible Markup Language (XML) 1.0*, Fourth Edition. World Wide Web Consortium. 2006. http://www.w3.org/TR/2006/REC-xml-20060816/

XML Namespaces, Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin (editors). *Namespaces in XML 1.1 (Second Edition)*. World Wide Web Consortium. 2006. http://www.w3.org/TR/2006/REC-xml-names11-20060816/

XML Schema Part 0: Primer (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-2/

(back to table)

CZ-03:

Page 7 section 1.2 states that UOML is expressed "with standard XML". However the submission does not contain any normative part with definition of XML format used by UOML. Each standard which builds on top of the XML should contain normative schema (ideally expressed using one or more languages from DSDL family – ISO/IEC 19757) – there is no such schema in the submission.

Section 2.1 contains definition of objects used by UOML abstract data model. However the definition of objects and properties is insufficient. There are no datatypes specified for properties. Definitions look very informal. The submission should use some widely recognized notation for specifying objects and their properties, for example UML or IDL.

The XML language and XML Schema are both noted as normative references within the specification. And the normative UOML XML schema does include data types for the UOML objects. http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd. A UML based figure has been added to the specification describing the abstract document model.

The following is a UML diagram of the UOML abstract document model. It shows the tree structure of UOML along with cardinalities associated with the objects discussed in this clause.

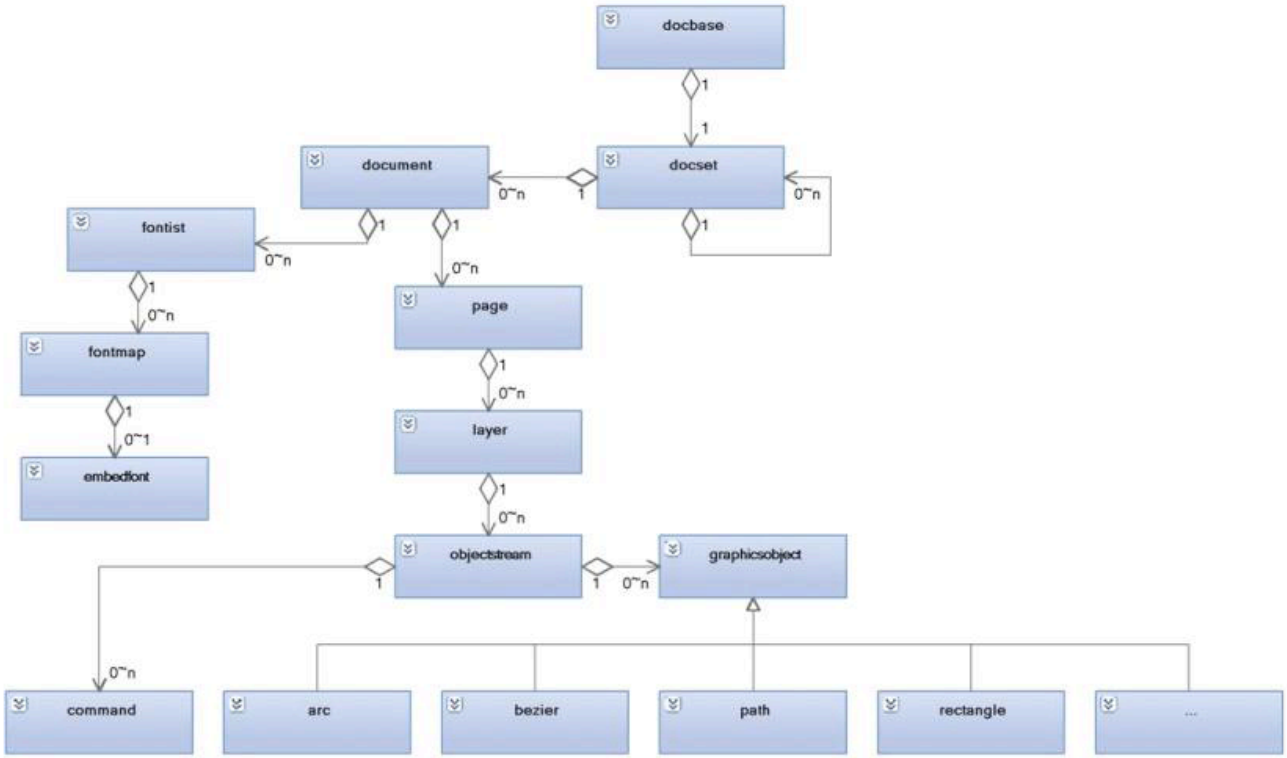


Figure 3. UML diagram of UOML abstract document model

(back to table)

CZ-05:
Section 2.3, 2.4, 2.5 and 2.6 contain definition of many properties for holding coordinates or dimensions of various objects. However nowhere is defined which measures can be used for specifying those.

Units and valid ranges will be defined for appropriate properties. And a new clause will be added discussing the general rules for data ranges.

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.
4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

Here is an example of a UOML object where valid data ranges have been added.

ELLIPSE

Semantics:

An ellipse, specified by a center, x and y radius, and a rotation angle.

Properties:

center: coordinates of ellipse center.

xr: positive integer value representing the length of the x-radius.

yr: positive integer value representing the length of the y-radius.

angle: inclination from coordinate system's x-axis to ellipse's x-axis. It is specified using a radian value. A positive value represents counterclockwise

and a negative value represents clockwise.

Sub-elements: N/A

(back to table)

CZ-06:

Section 3 defines operations on objects defined in the previous section which can be seen as definition of application interface. However the interface definition is underspecified and should use some common notation like IDL. Section 3 contains examples of XML fragments however it is not clear how these relate with operations defined. Moreover usage of XML namespace in example seems far from being uniform – some elements are namespaced, some are in no namespace. Naming conventions in presented XML fragments are not consistent – some elements use lowercase and some uppercase which is very inconvenient. There is no schema for elements used.

The UOML XML Schema is normative part of the UOML specification. All elements, data types are defined by this XSD. However, an informative annex was added showing detailed examples of how UOML could be used.

This annex is informative.

The examples below demonstrate the usage of many of the UOML instructions. Each example is followed by a corresponding “RET “instruction.

The XML string of a UOML instruction may be preceded by a prolog to specify the character encoding of the XML string. If default encoding is UTF-8, the prolog, `<?xml version="1.0" encoding="UTF-8"?>`, may be omitted. The default namespace for the XML string is: `urn:oasis:names:tc:uoml:xmlns:uoml:1.0`.

Example 1: open a docbase

```
Instructions sent from application to DCMS
<OPEN create="false" del_exist="false" path="c:\test.sep"/>
Instructions returned from DCMS to application
<!-- the string value “docbase001” is the opened docbase’s handle for later use -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docbase001"/>
</RET>
```

Example 2 : get the root docset of the docbase (following example 1)

```
Instructions sent from application to DCMS
<!-- since each docbase has one and only one sub-object, to get the root docset is just to get the first sub-object of docbase whose handle
is returned by example 1 -->
<GET handle="docbase001" usage="GET_SUB">
  <pos val="0"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docset001"/>
</RET>
```

Example 3: get the number of sub-objects of the root docset (following example 2)

```
Instructions sent from application to DCMS
<GET handle="docset001" usage="GET_SUB_COUNT"/>
Instructions returned from DCMS to application
<!-- the return value of 3 indicates the root docset has 3 sub-objects -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <intVal name="sub_count" val="3"/>
</RET>
```

Example 4: get the third sub-object of the docset (following example 3)

Instructions sent from application to DCMS

```
<GET handle="docset001" usage="GET_SUB">
  <pos val="2"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc001"/>
</RET>
```

Examples 5: get the type of a object using the empty string as the name of the property (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name=""/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="" val="DOC"/>
</RET>
```

Example 6: get the metadata of the document (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
```

```
  <property name="metainfo"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <compoundVal name="metainfo">
       <metalist>
            <meta key= "title" val="UOML Part I"/>
            <meta key="author" val="UOML TC"/>
       </metalist>
  </compoundVal>
</RET>
```

Example 7: get page bitmap of a page

```
Instructions sent from application to DCMS
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<GET usage="GET_PAGE_BMP" handle="page001">
  <disp_conf addr="c:\test.bmp" end_layer="8" format="bmp" output="FILE"
resolution="640">
    <clip>
      <ellipse angle="45" center="10,20" xr="30" yr="40"/>
      <roundrect br="70,80" tl="50,60" xr="90" yr="100"/>
      <subpath data="s 214,193 1 368,193 1 368,298 1 214,298"/>
```

```
    </clip>
  </disp_conf>
</GET>
Instructions returned from DCMS to application
<!-- the bmp format of page bitmap data has been saved in the file c:\test.bmp as requested -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 8 : get first layer of a page

```
Instructions sent from application to DCMS
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<!-- since page has only layer objects as its sub-objects, get sub-objects is the same to get layer objects -->
<GET  handle="page001" usage="GET_SUB">
  <pos val="0"/>
</GET>
Instructions returned from DCMS to application
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="layer001"/>
</RET>
```

Example 9: set a text object as the current object

```
Instructions send from application to DCMS
<!-- the text object's handle is supposed to have already obtained of value "text001" in prior instructions(using GET) -->
<USE  handle="text001"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Examples 10: get spaces property of a text object (following example 9)

```
Instructions send from application to DCMS
<GET  usage="GET_PROP">
  <property name="spaces"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="spaces" val="50,55"/>
</RET>
```

Example 11: insert a document into a docset (following example 2)

```
Instructions send from application to DCMS
<INSERT handle="docset001">
  <xobj>
```

```
    <doc name="UOML part II">
      <metainfo>
        <meta key="author" val="alex"/>
      </metainfo>
    </doc>
  </xobj>
</INSERT>
Instructions returned from DCMS to application
<!-- the handle of the inserted document is returned for later use  -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc002"/>
</RET>
```

Example 12: delete the document inserted in the example above

```
Instructions send from application to DCMS
<DELETE handle="doc002"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 13: use SYSTEM to save a docbase

```
Instructions send from application to DCMS
<SYSTEM>
  <flush path="c:\test.sep"/>
</SYSTEM>
<!-- instructions returned from DCMS to application -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 14: close the docbase (following example 1)

```
Instructions send from application to DCMS
<CLOSE  handle="docbase001"/>
instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

End of informative text.

(back to table)

DE-05:
"UOML provides detailed descriptions, explanations, examples, …" Improve the explanations in a way that they explain the purpose of UOML and the required behaviour of a DCMS

A detailed example will be added as an Appendix showing how UOML works with respect to the object model and instructions upon the model.

This annex is informative.

The examples below demonstrate the usage of many of the UOML instructions. Each example is followed by a corresponding “RET “instruction.

The XML string of a UOML instruction may be preceded by a prolog to specify the character encoding of the XML string. If default encoding is UTF-8, the prolog, `<?xml version="1.0" encoding="UTF-8"?>`, may be omitted. The default namespace for the XML string is:
`urn:oasis:names:tc:uoml:xmlns:uoml:1.0`.

Example 1: open a docbase

Instructions sent from application to DCMS

```
<OPEN create="false" del_exist="false" path="c:\test.sep"/>
```

Instructions returned from DCMS to application

```
<!-- the string value “docbase001” is the opened docbase’s handle for later use -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docbase001"/>
</RET>
```

Example 2 : get the root docset of the docbase (following example 1)

Instructions sent from application to DCMS

```
<!-- since each docbase has one and only one sub-object, to get the root docset is just to get the first sub-object of docbase whose handle is returned by example 1 -->
```

```
<GET handle="docbase001" usage="GET_SUB">
  <pos val="0"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docset001"/>
</RET>
```

Example 3: get the number of sub-objects of the root docset (following example 2)

```
Instructions sent from application to DCMS
<GET handle="docset001" usage="GET_SUB_COUNT"/>
Instructions returned from DCMS to application
<!-- the return value of 3 indicates the root docset has 3 sub-objects -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <intVal name="sub_count" val="3"/>
</RET>
```

Example 4: get the third sub-object of the docset (following example 3)

```
Instructions sent from application to DCMS
<GET handle="docset001" usage="GET_SUB">
  <pos val="2"/>
```

```
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc001"/>
</RET>
```

Examples 5: get the type of a object using the empty string as the name of the property (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name=""/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="" val="DOC"/>
</RET>
```

Example 6: get the metadata of the document (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name="metainfo"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <compoundVal name="metainfo">
       <metalist>
             <meta key= "title" val="UOML Part I"/>
             <meta key="author" val="UOML TC"/>
       </metalist>
  </compoundVal>
</RET>
```

Example 7: get page bitmap of a page

```
Instructions sent from application to DCMS
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<GET usage="GET_PAGE_BMP" handle="page001">
  <disp_conf addr="c:\test.bmp" end_layer="8" format="bmp" output="FILE"
resolution="640">
    <clip>
      <ellipse angle="45" center="10,20" xr="30" yr="40"/>
      <roundrect br="70,80" tl="50,60" xr="90" yr="100"/>
      <subpath data="s 214,193 1 368,193 1 368,298 1 214,298"/>
    </clip>
  </disp_conf>
</GET>
```

Instructions returned from DCMS to application

```
<!-- the bmp format of page bitmap data has been saved in the file c:\test.bmp as requested -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 8 : get first layer of a page

Instructions sent from application to DCMS

```
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<!-- since page has only layer objects as its sub-objects, get sub-objects is the same to get layer objects -->
<GET  handle="page001" usage="GET_SUB">
  <pos val="0"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="layer001"/>
</RET>
```

Example 9: set a text object as the current object

Instructions send from application to DCMS

```
<!-- the text object's handle is supposed to have already obtained of value "text001" in prior instructions(using GET) -->
<USE  handle="text001"/>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Examples 10: get spaces property of a text object (following example 9)

Instructions send from application to DCMS

```
<GET  usage="GET_PROP">
  <property name="spaces"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="spaces" val="50,55"/>
</RET>
```

Example 11: insert a document into a docset (following example 2)

Instructions send from application to DCMS

```
<INSERT handle="docset001">
  <xobj>
    <doc name="UOML part II">
      <metainfo>
        <meta key="author" val="alex"/>
```

```
      </metainfo>
    </doc>
  </xobj>
</INSERT>
Instructions returned from DCMS to application
<!-- the handle of the inserted document is returned for later use  -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc002"/>
</RET>
```

Example 12: delete the document inserted in the example above

```
Instructions send from application to DCMS
<DELETE handle="doc002"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 13: use SYSTEM to save a docbase

```
Instructions send from application to DCMS
<SYSTEM>
  <flush path="c:\test.sep"/>
```

```
</SYSTEM>
<!-- instructions returned from DCMS to application -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 14: close the docbase (following example 1)

Instructions send from application to DCMS

```
<CLOSE  handle="docbase001"/>
```

instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

End of informative text.

(back to table)

DE-07:
The feature set offered by UOML is too small and not generic enough to fulfill the promised interoperability of daily used documents or the promise given on interoperability.

Provided wording in the Overview stating the possible interoperability with other formats, e.g., PDF

The UOML graphics object model is similar to the graphics model specified by ISO/IEC 32000-1:2008, the Portable Document Format (PDF) standard. For example, both standards describe a page layout using logical coordinate systems, and the positions of the graphics objects are specified using coordinates in the logical coordinate systems. The similarity of the two models allows UOML to be used as an interface standard for PDF.

(back to table)

DE-08:
The feature set offered by UOML is too small and not generic enough to fulfill the promised interoperability of daily used documents or the promise given on interoperability.

More terms have been added to the Terminology clause and existing definitions have been clarified as needed. Also, within the specification further clarification has been added during changes based on NB comments.

application software: Software that handles document data by issuing UOML instructions to the DCMS.

command object: An object used to modify the graphics state that holds the current graphics control parameters, such as text size, typeface and color.

DCMS: Abbreviated for “Docbase Management System”.

docbase: The root level of the UOML abstract document model. Abbreviated for “document base”, it is the container of one or many documents. A docbase contains one and only one root docset. [*Note*: The docbase is analogous to a file system on a modern operating system. The term docbase is derived from the term “database”. The docset is analogous to a directory within a file system on a modern operating system. The root docset is analogous to the root directory of a file system. *end note*].

Docbase Management System: The software that implements the functionality defined by the UOML specification. Abbreviated as DCMS.

docset: A set of documents. A docset may contain one to many docsets. [*Note*: The docset is analogous to a directory within a file system on a modern operating system. *end note*].

document global object: A document global object may include a fontlist, fontmap and/or embedfont.

graphics object: An object that is drawable by the render engine. It describes part or all of the appearance on a page. Examples include images and text.

graphics state: An internal structure maintained by the DCMS to hold current graphics control parameters. A command object changes one or multiple parameters in the current graphics state.

graphics state stack: A sequence of graphics states where the first one in is the last one out. A DCMS shall maintain a stack for graphics states, called the graphics state stack**.** [*Note*: The command object PUSH_GS saves a copy of the current graphics state onto the stack. The command object POP_GS restores the saved copy, remove it from the stack and make it the current graphics state. *end note*]

layer: A page is composed of one or more layers. A layer has the same size as the page on which it is constructed. The visual appearance of a page is a combination of all of the layers of the page.

object: The UOML abstract document model is a tree structure, and a node in the tree is called a UOML object, abbreviated as object.

object stream: A sequence of graphics objects and command objects. A layer holds object streams.

page bitmap: A raster image that represents the visual appearance of the page. The number of pixels of the raster image depends on the resolution of the raster image. The number of pixels in the horizontal direction equals the page width multiplied by the resolution; the number of pixels in the vertical direction equals the page height multiplied by the resolution. [*Note*: The resolution is the same for both the horizontal and vertical direction. *end note*]

Path: A Path is a graphics object composed of straight and/or curved line segments, which may or may not be connected. [*Note*: that in this document, 'path' (all lowercase) refers to a filename, location of docbase or image file. This is different from this current definition of “Path” (with the uppercase ‘P’). *end note*]

sub-element: In a UOML object XML representation, a sub-element is the child XML node of its parent XML node.

sub-object: In the UOML abstract document model tree structure instance, a sub-object is the child node of its parent object node. Each sub-objet has only one parent node. A parent node may have multiple sub-objects as child nodes. [*Note*: A Sub-object is created by the UOML INSERT instruction. *end note*]

UOML: abbreviation of "Unstructured Operation Markup Language".

(back to table)

DE-09:
The relationship between a Docbase and the root Docset is unclear. One seems obsolete.

Clarified the relationship between Docbase and Docset in the Terminology clause.

docbase: The root level of the UOML abstract document model. Abbreviated for “document base”, it is the container of one or many documents. A docbase contains one and only one root docset. [*Note*: The docbase is analogous to a file system on a modern operating system. The term docbase is derived from the term “database”. The docset is analogous to a directory within a file system on a modern operating system. The root docset is analogous to the root directory of a file system. *end note*].

docset: A set of documents. A docset may contain one to many docsets. [*Note*: The docset is analogous to a directory within a file system on a modern operating system. *end note*].

Also added some wording in the Abstract Document Model.

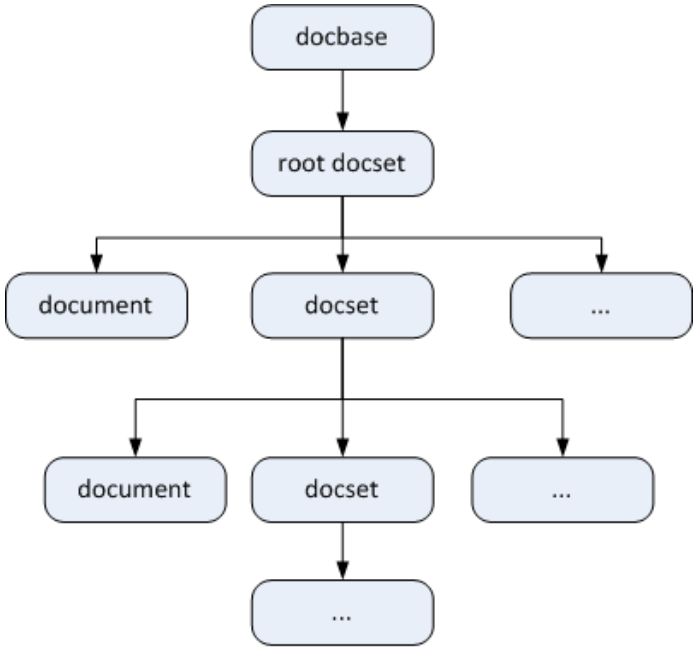


Figure1. UOML Abstract document Model 1

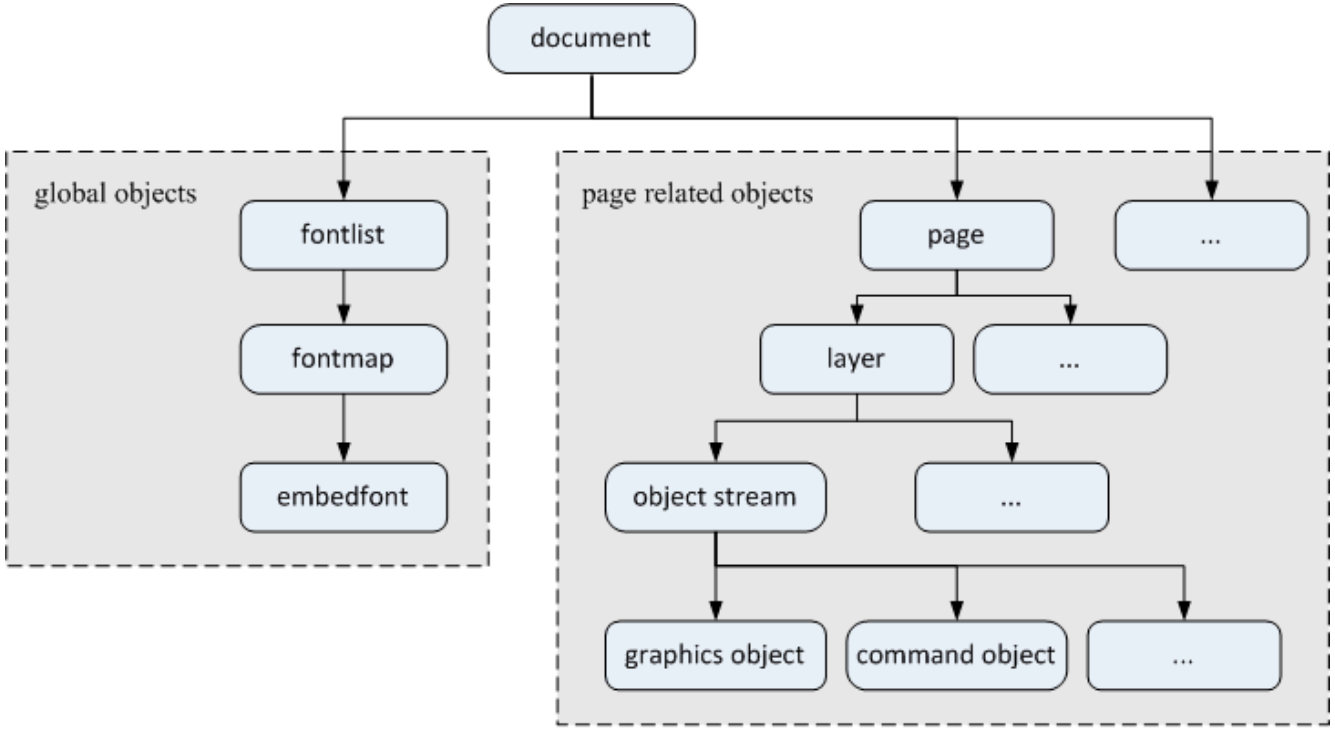


Figure 2. UOML Abstract Document Model 2

Docbase

The docbase is the root of the UOML abstract document model structure. A docbase has only one docset sub-object called the root docset [*Note*: Other docsets and documents are a docset's sub-objects. *end note*].

The root docset is generated automatically when the docbase is created (see Figure 1). In this specification, the docbase object is specified using DOCBASE (§4.3).

Sub-object: docset

(back to table)

DE-12:

We are missing definitions for the following terms that appear to have specific meanings in this specification:

- **sub-element**
- **direct sub-object**
- **indirect sub-object**

Added a definition for sub-element in Terminology. Also removed references to direct and indirect subobject.

sub-element: In a UOML object XML representation, a sub-element is the child XML node of its parent XML node.

(back to table)

DE-14:
This Clause states that "Page Data may include 0 ... page(s)". We do not know document with no pages!? What about Global Date like i.e. Metadata?

Added clarifying wording in the Internal Structure clause describing why 0 pages are permitted.

Document

The document object is the root node of document information (see Figure 2). A document contains static information for fixed-layout 2D documents [*Note*: In future UOML parts or future versions of this part, other types of document information may be supported, including audio/video, 3D information, etc. *end note*]. A single document has zero to multiple pages. In this specification, a document object is specified using DOC. (§4.5).

[*Note*: A document with no pages is permitted. It is an intermediate state. One can create such a document, then open and add pages at a future time. *end note*]

Sub-object: fontlist, page.

(back to table)

DE-16:
The properties width, height, resolution are underspecified. What are the possible units?

Provided specific units for width, height and resolution (e.g. logical pixels).

PAGE

Semantics: XML description of the page object (§2.6).

Properties:

width: positive float value representing the width of the page in pixels.

height: positive float value representing the height of the page in pixels.

resolution: positive integer value representing the resolution of the page, which defines the unit of a pixel (§4.1).

Sub-elements: N/A

Also added a clause regarding the logical coordinate system and units

Logical Coordinate System and Units

A UOML document uses a logical coordinate system. The terms *position*, *point* and *coordinate* may be used interchangeably. They refer to a logical point in the logical coordinate system. The origin of the logical coordinate system is the top left point. The direction of the x-axis is left to right. The direction of the y-axis is top to bottom.

The length of the units along each axis depends on the resolution property of the page. If the resolution of a page is x, the length of the unit along each axis is 1/x inch. A logical unit indicates one inch divided by the resolution of the page.

The resolution of each page is the same along the x and y axis.

UOML uses radians as the unit of measurement for angles. [*Note*: Though different from PDF, XSL-FO and SVG, conversion can be easily made without any loss of information. *end note*]

(back to table)

DE-20:
Missing normative reference in 1.3 to a specification for the image file formats 'BMP' or 'TIFF', actually these are referred in the normative text.

Related normative references have been added.

[BMP] Bitmap Format. BMP. http://msdn.microsoft.com/en-us/library/at62haz6.aspx

[TIFF] ISO 12639:2004, *Graphic technology — Prepress digital data exchange — Tag image file format for image technology (TIFF/IT)*.

(back to table)

DE-21:
It is unclear how the spaces property is to be interpreted. Interpretation of all properties is missing.

While all data types are normatively in the UOML XML Schema definition (XSD), additional wording has been added to the TEXT object regarding the spaces property.

TEXT

Semantics:

Text, specified using an origin, encoding information, text data and an optional character spacing list.

Properties:

origin: the coordinate of the first character's origin. The origin of a character is defined by its font information.

encode: character set or encoding of text data. The valid value for this property should be one of the character encodings registered (as charsets) with the Internet Assigned Numbers Authority [IANA-CHARSETS], otherwise it should use names starting with an x- prefix.

text: character data contained in text, base64 encoded string data.

spaces: an optional, ordered set of distances that specifies distances between adjacent characters' origins, separated by a comma.

The origin of a character refers to the point (0, 0) in the coordinate system of the character glyph, as illustrated in the Figure 4. When a text object with only one character is specified and the text object has coordinate (x, y), the rendering engine should place the origin of the character at (x, y) and render the character.

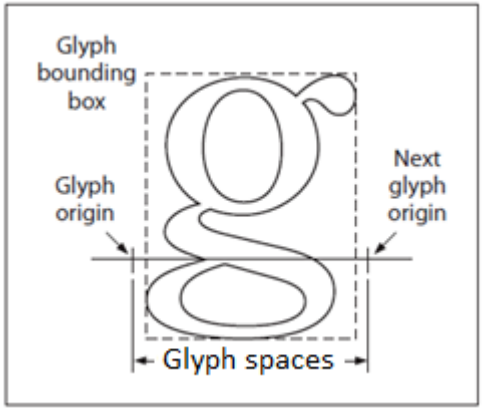


Figure 4. spaces of text

The spaces property is the offset or distance between the x coordinates of two adjacent characters. It is always positive. The number of comma-separated values shall be one fewer than the number of characters in the string. The values should override the widths of the characters as specified by the font used. The values are used to calculate the coordinate to place the origin of each character.

Sub-elements: N/A

Also, an example has been added to show how UOML might function, including an example using Spaces.

Example 9: set a text object as the current object

```
Instructions send from application to DCMS
<!-- the text object's handle is supposed to have already obtained of value "text001" in prior instructions(using GET) -->
<USE  handle="text001"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Examples 10: get spaces property of a text object (following example 9)

```
Instructions send from application to DCMS
<GET  usage="GET_PROP">
  <property name="spaces"/>
</GET>
Instructions returned from DCMS to application
```

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="spaces" val="50,55"/>
</RET>
```

(back to table)

DE-23:

Reference to the supported color model is missing.

Added normative reference to RGB in the Normative References clause.

[RGB] IEC 61966-2-1: 1999: Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB. International Electrotechnical Commission, 1999. ISBN 2-8318-4989-6 as amended by Amendment A1:2003.

(back to table)

DE-25:

Missing definition, specification, relationships, properties, values or value range of POP_GS, PUSH_GS, CHAR_STYLE, SHADOW_WIDTH, SHADOW_LEN, SHADOW_DIR, CLIP_AREA

All of these elements are specified within the UOML specification.

CHAR_STYLE

Semantics: Set the current character style.

Properties:

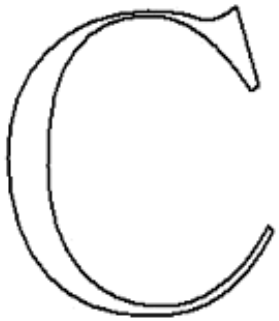
v1: a character string, representing the character style. The possible values for this property are SHADOW, HOLLOW and OUTLINE, or some combination of the three, separated by commas. If the string is set to empty, then any previous setting is cleared.

SHADOW: set shadow style. If this character style is set, then the following algorithm is used to render the shadow effect:

- If SHADOW_NEG (§4.11.2.30) is false, the character is extended with a distance of SHADOW_LEN (§4.11.2.27) along the shadow direction (§4.11.2.28), then a hollowed character with raster operation ROP_COPY is drawn in the original position. The border width of the hollowed character is SHADOW_WIDTH (§4.11.2.26).

- If SHADOW_NEG is true, the character position is moved with a distance of SHADOW_LEN along the shadow direction, and extended SHADOW_WIDTH along the shadow direction; then the character is drawn in the original position with background color and raster operation ROP_COPY, and extended with a distance SHADOW_LEN along the shadow direction; then in the original position, a character with normal color and raster operation ROP_COPY is drawn.

HOLLOW: set hollow style. If this character style is set, a line with thickness HOLLOW_BORDER (§4.11.2.35) should be drawn along the outline of the character.



OUTLINE: set outline style. If this character style is set, a line with thickness OUTLINE_BORDER (§4.11.2.33), and with distance OUTLINE_WIDTH (§4.11.2.34) from the outline of the character, should be drawn along the outline of the character.

Sub-elements: N/A

PUSH_GS

Semantics: Push the current graphics state onto the graphics state stack.

Properties: N/A

Sub-elements: N/A

POP_GS

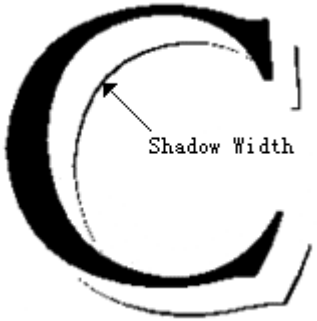
Semantics: Pop out the top value from the graphics state stack, replacing current graphics state.

Properties: N/A

Sub-elements: N/A

SHADOW_WIDTH

Semantics: Set the border width of the current character shadow. SHADOW_WIDTH represents the thickness of the outline of a shadow.



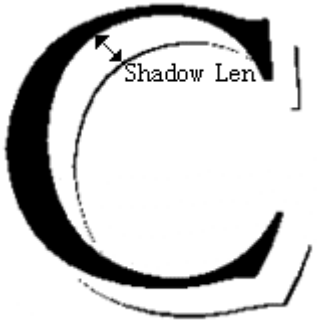
Properties:

v1: a non-negative floating point number, representing the shadow border width.

Sub-elements: N/A

SHADOW_LEN

Semantics: Set the length of the current character shadow. SHADOW_LEN represents the distance of the shadow from the corresponding point of the character along the character shadow direction (§4.11.2.28) to the edge of the shadow width.



Properties:

v1: a non-negative floating point number, representing the character shadow length.

Sub-elements: N/A

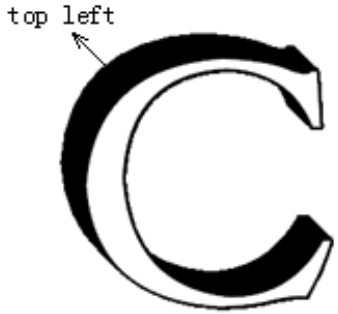
SHADOW_DIR

Semantics: Set the direction of the current character shadow

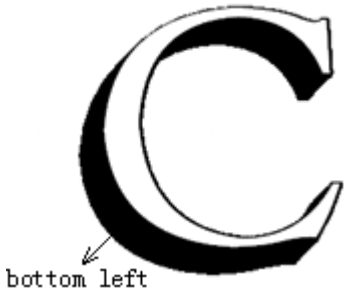
Properties:

v1: a character string. The possible values for this property are SHADOW_LT, SHADOW_LB, SHADOW_RT and SHADOW_RB.

SHADOW_LT: the character shadow direction is top left.



SHADOW_LB: the character shadow direction is bottom left.



SHADOW_RT: the character shadow direction is top right.

SHADOW_RB: the character shadow direction is bottom right.

Sub-elements: N/A

CLIP_AREA

Semantics: Set the current clip area

Properties: N/A

Sub-elements:

cliparea: PATH type, representing the new clip area.

The Path specified by a CLIP_AREA command object is relative to the page. The portions of graphic objects that lie outside of the current clip area are not rendered.

A 'Valid Ranges' clause has been added to specify the general valid values of data types.

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.
4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

(back to table)

DE-28:
No dependencies between operations available.

Described the order of operations in the UOML instructions clause.

This clause defines the syntax and semantics of the UOML instructions. The order of UOML instructions are OPEN, followed by zero or many operations except OPEN or CLOSE, ended by CLOSE. There are no dependencies among operations between OPEN and CLOSE; thus there is no order for those operations.

(back to table)

DE-30:
The conformance statement can't be validated.

The Conformance clause has been re-worded to clarify UOML validation.

The text in this OASIS standard is divided into *normative* and *informative* categories. Unless documented otherwise, all features specified in normative text of this OASIS standard shall be implemented. Text marked informative (using the mechanisms described in §1.5) is for information purposes only. Unless stated otherwise, all text is normative.

Use of the word "shall" indicates required behavior.

Any behavior that is not explicitly specified by this OASIS standard is implicitly unspecified (§4).

DCMS Conformance

A UOML Document Management System (DCMS) has conformance if it implements all of the UOML instructions in compliance with the syntax and semantics in this OASIS standard.

Application Conformance

A UOML application is conformant if both of the following are true:

- The application issues UOML instructions to the DCMA as specified in this OASIS standard; and
- The application parses the return instructions from the DCMS according to this OASIS standard.

(back to table)

FR-03:
This document is very incomplete on many aspects:

- Almost no units or data types are provided (width, height of page? resolution? tl and br coordinates, etc.)
- No information is provided for allowed values (what does it mean to have a width with a negative value)

The specification has been updated in various clauses to clarify units and proper values, including a Data Ranges and Logical Coordinate System and Units clause. Data types are defined, however, in the normative XML Schema definition, located at: http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd. Here is an example for Page:

PAGE

Semantics: XML description of the page object (§2.6).

Properties:

width: positive float value representing the width of the page in pixels.

height: positive float value representing the height of the page in pixels.

resolution: positive integer value representing the resolution of the page, which defines the unit of a pixel (§4.1).

Sub-elements: N/A

Here is the Data Ranges sub-clause:

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.
4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

Here is the Logical Coordinate System and Units sub-clause:

Logical Coordinate System and Units

A UOML document uses a logical coordinate system. The terms *position*, *point* and *coordinate* may be used interchangeably. They refer to a logical point in the logical coordinate system. The origin of the logical coordinate system is the top left point. The direction of the x-axis is left to right. The direction of the y-axis is top to bottom.

The length of the units along each axis depends on the resolution property of the page. If the resolution of a page is x, the length of the unit along each axis is 1/x inch. A logical unit indicates one inch divided by the resolution of the page.

The resolution of each page is the same along the x and y axis.

UOML uses radians as the unit of measurement for angles. [*Note*: Though different from PDF, XSL-FO and SVG, conversion can be easily made without any loss of information. *end note*]

(back to table)

FR-04:
It is very hard to understand when the API starts and where the Format stops
*** What is the definition of a sub-element?**
*** What is a sub-object?**
*** What is the relationship between those two?**

In the Terminology clause, the following wording was added or changed:

sub-element: In a UOML object XML representation, a sub-element is the child XML node of its parent XML node.

sub-object: In the UOML abstract document model tree structure instance, a sub-object is the child node of its parent object node. Each sub-objet has only one parent node. A parent node may have multiple sub-objects as child nodes. [*Note*: A Sub-object is created by the UOML INSERT instruction. *end note*]

In the Abstract Document Model and UOML Instructions clause, the following informative notes were added:

[*Note*:

A sub-object describes part of the logical model of the UOML object tree. For example, a logical model of a document could be:

```
docbase
  docset
    document
       page
         layer
            object stream
```

where the child object is the sub-object of the parent object. For example, `document` is the sub-object of `docset`, `page` is the sub-object of `doc`, etc. However, there is no XML representation of the UOML docbase since UOML does not specify the format of document. The XML schema of each UOML describes the object itself.

end note]

[*Note*:

In UOML a sub-element is a child XML element in UOML object's XML representation. For example, the XML representation of a CMD object in UOML could be:

```
<CMD name="COLOR_LINE" >
<rgb r="128" g="3" b="255" a="120"/>
</CMD>
```

where `rgb` is a sub element of `cmd`.

end note]

(back to table)

FR-05:
The RFC 2119 (about the correct use of SHOULD MAY MUST, etc.) is not used at all: It should be used
*** What are the levels of conformance?**

Added a complete conformance clause to the specification:

The text in this OASIS standard is divided into *normative* and *informative* categories. Unless documented otherwise, all features specified in normative text of this OASIS standard shall be implemented. Text marked informative (using the mechanisms described in §1.5) is for information purposes only. Unless stated otherwise, all text is normative.

Use of the word “shall” indicates required behavior.

Any behavior that is not explicitly specified by this OASIS standard is implicitly unspecified (§4).

DCMS Conformance

A UOML Document Management System (DCMS) has conformance if it implements all of the UOML instructions in compliance with the syntax and semantics in this OASIS standard.

Application Conformance

A UOML application is conformant if both of the following are true:

The application issues UOML instructions to the DCMA as specified in this OASIS standard; and
The application parses the return instructions from the DCMS according to this OASIS standard.

(back to table)

FR-06:
The document is entitled First part:
*** Why the first part?**
*** What are the other parts about?**
*** Why is it split into parts?**
*** Are we voting on a multipart standard, by seeing only the first part?**

Added wording to the introductory Overview explaining why this is Part 1 of the spec:

This OASIS standard forms the foundation of UOML. Other standards building upon this standard may be created in the future.

(back to table)

GB-02:

It is stated in the Explanatory Report accompanying the specification that: "There are two parts of software involved in the use of the UOML standard, one is the application software that consumes or displays documents; the other is a DCMS (DoCbase Management System)". How is a DCMS to be implemented? What specifications and IPR regimes govern DCMS implementations?

Here is an example, simple implementation of a DCMS.

The docbase data in a DCMS can be described in XML and when the abstract document model is hierarchical, an XML tree can be built accordingly. An operation of inserting adds a node in the XML tree; an operation of deleting deletes a node in the XML tree; an operation of setting sets the attributes of a corresponding node or the child node of a corresponding node; an operation of getting gets the attributes of the corresponding node and returns the attribute information to the application. A further description of an embodiment is given as follows:

1. XML is used for describing every object; therefore an XML tree is created for each object. Some objects show simple attributes and the XML trees corresponding to the objects will have only the root node; some objects show complicated attributes and the XML trees corresponding to the objects will have a root node and child nodes. The description of the XML trees can be created with reference to the XML definitions of the operation objects provided in the UOML specification.
2. When a new docbase is created, a new XML file whose root node is the docbase object is created.
3. When a new object (e.g., a text object) is inserted into the docbase, the XML tree corresponding to the new object is inserted under the corresponding parent node (e.g., a layer). Therefore, every object in the docbase corresponds to a node in the XML tree whose root node is the docbase.
4. When an object is deleted, the node corresponding to the object and the child nodes thereof are deleted. The deletion starts from a leaf node in a tree traversal from the bottom to the top.
5. When an attribute of an object is set, the attribute of the node corresponding to the object is set to the same value. If the attribute is expressed as an attribute of a child node, the attribute of the corresponding childe node is set to the same value.
6. In the process of getting an attribute of an object, the node corresponding to the object is accessed and the attribute of the object is retrieved according to the corresponding attribute and child nodes of the node.
7. In the process of getting the handle of an object, the XML path of the node corresponding to the object is returned.
8. When an object (e.g., a page) is copied to a specified position, the whole subtree starting from the node corresponding to the object is copied to a position right under the parent node corresponding to the specified position (e.g., a document). When the object is copied to another docbase, the object referenced by the subtree (e.g., an embedded font) is also copied.
9. In the process of performing an instruction of getting a page bitmap, a blank bitmap in a specified bitmap format is created first in the same size of the specified area, then all layout objects of the specified page are traversed. Every layout object in the specified area (including the objects that

have only parts in the area) is rendered and displayed in the blank bitmap. The process is complicated and can be performed by those skilled in the art; however, the process is still covered by the RIP (Raster ImageProcessor) technology in the prior art and will not be described herein.

(back to table)

GB-03:

The specification provides no information on the business or use case that UOML is intended to meet. It would appear that all existing implementations of this specification have been in connection with use of a particular DCMS implementation by the company Sursen. It does not seem to us that this constitutes a sufficient case for proposing an International Standard. We have been unable to identify any use cases for this specification within the United Kingdom.

Added a business case for the existence of a UOML specification in the Overview sub-clause.

The UOML graphics object model is similar to the graphics model specified by ISO/IEC 32000-1:2008, the Portable Document Format (PDF) standard. For example, both standards describe a page layout using logical coordinate systems, and the positions of the graphics objects are specified using coordinates in the logical coordinate systems. The similarity of the two models allows UOML to be used as an interface standard for PDF.

Here is a more detailed business case

.

The project "Electronic Document Workflow Platform Based on UOML" is used in the State Electricity Regulatory Commission of China, and has achieved good results. The State Electricity Regulatory Commission (SERC) is empowered by the State Council to perform administrative and regulatory duties with regard to the national electric power sector (in accordance with laws and regulations). It is responsible for the overall regulation of the national power sector, establishing a coherent system for regulatory organization, and exercising direct leadership over its regional branches.

The project "Electronic document workflow platform Based on UOML" in SERC integrated various UOML- conforming software products, including SURSEN electronic document transmission system, HANVON OCR software, the REDFLAG office suite software, the JETRICH handwriting software, FARSHIEN barcode identification system, TRS full-text indexing and UNIS archive management system. All these software products can open UOML-accessible documents, appending various elements, such as text, handwriting signature and two-dimensional barcodes. These products can also perform full-text search on UOML-accessible documents. As the products are all based on unified UOML interface, the document can interoperate among different software, so that the goal of a truly "paperless offices" is achieved finally with the office working environment quite efficient and convenient.

It is shown via this successful user case that several UOML-conformed products already exist from different vendors, and these products have been formally used in key applications.

The demonstration of this UOML user case can be found: http://www.uoml.org/flv04.html

(back to table)

GB-04:
The specification does not include a clear statement its Scope. There is an informal description in Clause 1.2, but this doesn't provide what is needed.

A Scope clause has been added to the UOML specification.

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

(back to table)

GB-05:
The specification is labelled as being Part 1 of, presumably, a multi-Part specification. Yet no information is provided about any other Parts, or about the dependencies between them.

Added wording to the introductory Overview explaining why this is Part 1 of the spec:

This OASIS standard forms the foundation of UOML. Other standards building upon this standard may be created in the future.

(back to table)

GB-06:

The specification makes normative reference to several standards and reference specifications, and also makes a non-normative reference to ISO/IEC FDIS 32000. However there is no explanation of how this specification relates to ISO/IEC FDIS 32000, nor to a number of other important standards used in layout-oriented representations of documents (e.g. ISO 10180 SPDL, XSL-FO, SVG).

Added an explanation in the Overview of how PDF plays a role with UOML:

The UOML graphics object model is similar to the graphics model specified by ISO/IEC 32000-1:2008, the Portable Document Format (PDF) standard. For example, both standards describe a page layout using logical coordinate systems, and the positions of the graphics objects are specified using coordinates in the logical coordinate systems. The similarity of the two models allows UOML to be used as an interface standard for PDF.

(back to table)

GB-07:

The specification as a whole is lacking in detail, making it frequently difficult to interpret what is meant. The Clause-specific comments below cover the worst cases that have been found, but there is a general sense in which this specification has not been editing for clarity. While it is no doubt self-evident to the authors what is meant, given their familiarity with the concepts and their experience with existing implementations, it should not be assumed that someone unfamiliar with the technology will start with the same level of conceptual understanding.

The text has been edited for grammatical clarity and an expanded Terminology clause now exists:

For the purposes of this document, the following terms and definitions apply. Other terms are defined where they appear in *italics* typeface. Terms not explicitly defined in this Part of this OASIS standard are not to be presumed to refer implicitly to similar terms defined elsewhere.

Throughout this Part of this OASIS standard, the terminology “must”, “must not”, “required”, “shall”, “shall not”, “should”, “should not”, “recommended”, “may” and “optional” in this document shall be interpreted as described in RFC 2119, *Keywords for use in RFCs to Indicate Requirement Levels*. [RFC2119].

application software: Software that handles document data by issuing UOML instructions to the DCMS.

command object: An object used to modify the graphics state that holds the current graphics control parameters, such as text size, typeface and color.

DCMS: Abbreviated for “Docbase Management System”.

docbase: The root level of the UOML abstract document model. Abbreviated for “document base”, it is the container of one or many documents. A docbase contains one and only one root docset. [*Note*: The docbase is analogous to a file system on a modern operating system. The term docbase is derived from the term “database”. The docset is analogous to a directory within a file system on a modern operating system. The root docset is analogous to the root directory of a file system. *end note*].

Docbase Management System: The software that implements the functionality defined by the UOML specification. Abbreviated as DCMS.

docset: A set of documents. A docset may contain one to many docsets. [*Note*: The docset is analogous to a directory within a file system on a modern operating system. *end note*].

document global object: A document global object may include a fontlist, fontmap and/or embedfont.

graphics object: An object that is drawable by the render engine. It describes part or all of the appearance on a page. Examples include images and text.

graphics state: An internal structure maintained by the DCMS to hold current graphics control parameters. A command object changes one or multiple parameters in the current graphics state.

graphics state stack: A sequence of graphics states where the first one in is the last one out. A DCMS shall maintain a stack for graphics states, called the graphics state stack**.** [*Note*: The command object PUSH_GS saves a copy of the current graphics state onto the stack. The command object POP_GS restores the saved copy, remove it from the stack and make it the current graphics state. *end note*]

layer: A page is composed of one or more layers. A layer has the same size as the page on which it is constructed. The visual appearance of a page is a combination of all of the layers of the page.

object: The UOML abstract document model is a tree structure, and a node in the tree is called a UOML object, abbreviated as object.

object stream: A sequence of graphics objects and command objects. A layer holds object streams.

page bitmap: A raster image that represents the visual appearance of the page. The number of pixels of the raster image depends on the resolution of the raster image. The number of pixels in the horizontal direction equals the page width multiplied by the resolution; the number of pixels in the vertical direction equals the page height multiplied by the resolution. [*Note*: The resolution is the same for both the horizontal and vertical direction. *end note*]

Path: A Path is a graphics object composed of straight and/or curved line segments, which may or may not be connected. [*Note*: that in this document, 'path' (all lowercase) refers to a filename, location of docbase or image file. This is different from this current definition of “Path” (with the uppercase ‘P’). *end note*]

sub-element: In a UOML object XML representation, a sub-element is the child XML node of its parent XML node.

sub-object: In the UOML abstract document model tree structure instance, a sub-object is the child node of its parent object node. Each sub-objet has only one parent node. A parent node may have multiple sub-objects as child nodes. [*Note*: A Sub-object is created by the UOML INSERT instruction. *end note*]

UOML: abbreviation of "Unstructured Operation Markup Language".

(back to table)

GB-09:
In general the definitions provided in this specification are too terse and therefore liable to misinterpretation.

More detail was added to the Terminology clause, along with a detailed example in an Annex:

This annex is informative.

The examples below demonstrate the usage of many of the UOML instructions. Each example is followed by a corresponding “RET “instruction.

The XML string of a UOML instruction may be preceded by a prolog to specify the character encoding of the XML string. If default encoding is UTF-8, the prolog, `<?xml version="1.0" encoding="UTF-8"?>`, may be omitted. The default namespace for the XML string is: `urn:oasis:names:tc:uoml:xmlns:uoml:1.0.`

Example 1: open a docbase

```
Instructions sent from application to DCMS

<OPEN create="false" del_exist="false" path="c:\test.sep"/>

Instructions returned from DCMS to application

<!-- the string value “docbase001” is the opened docbase’s handle for later use -->

<RET>

  <boolVal name="SUCCESS" val="true"/>

  <stringVal name="handle" val="docbase001"/>

</RET>
```

Example 2 : get the root docset of the docbase (following example 1)

```
Instructions sent from application to DCMS

<!-- since each docbase has one and only one sub-object, to get the root docset is just to get the first sub-object of docbase whose handle
is returned by example 1 -->
```

```
<GET handle="docbase001" usage="GET_SUB">
  <pos val="0"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docset001"/>
</RET>
```

Example 3: get the number of sub-objects of the root docset (following example 2)

```
Instructions sent from application to DCMS
<GET handle="docset001" usage="GET_SUB_COUNT"/>
Instructions returned from DCMS to application
<!-- the return value of 3 indicates the root docset has 3 sub-objects -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <intVal name="sub_count" val="3"/>
</RET>
```

Example 4: get the third sub-object of the docset (following example 3)

```
Instructions sent from application to DCMS
<GET handle="docset001" usage="GET_SUB">
  <pos val="2"/>
```

```
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc001"/>
</RET>
```

Examples 5: get the type of a object using the empty string as the name of the property (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name=""/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="" val="DOC"/>
</RET>
```

Example 6: get the metadata of the document (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name="metainfo"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <compoundVal name="metainfo">
       <metalist>
             <meta key= "title" val="UOML Part I"/>
             <meta key="author" val="UOML TC"/>
       </metalist>
  </compoundVal>
</RET>
```

Example 7: get page bitmap of a page

```
Instructions sent from application to DCMS
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<GET usage="GET_PAGE_BMP" handle="page001">
  <disp_conf addr="c:\test.bmp" end_layer="8" format="bmp" output="FILE"
resolution="640">
    <clip>
      <ellipse angle="45" center="10,20" xr="30" yr="40"/>
      <roundrect br="70,80" tl="50,60" xr="90" yr="100"/>
      <subpath data="s 214,193 1 368,193 1 368,298 1 214,298"/>
    </clip>
  </disp_conf>
</GET>
```

Instructions returned from DCMS to application

```
<!-- the bmp format of page bitmap data has been saved in the file c:\test.bmp as requested -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 8 : get first layer of a page

Instructions sent from application to DCMS

```
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<!-- since page has only layer objects as its sub-objects, get sub-objects is the same to get layer objects -->
<GET  handle="page001" usage="GET_SUB">
  <pos val="0"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="layer001"/>
</RET>
```

Example 9: set a text object as the current object

Instructions send from application to DCMS

```
<!-- the text object's handle is supposed to have already obtained of value "text001" in prior instructions(using GET) -->
<USE  handle="text001"/>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Examples 10: get spaces property of a text object (following example 9)

Instructions send from application to DCMS

```
<GET  usage="GET_PROP">
  <property name="spaces"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="spaces" val="50,55"/>
</RET>
```

Example 11: insert a document into a docset (following example 2)

Instructions send from application to DCMS

```
<INSERT handle="docset001">
  <xobj>
    <doc name="UOML part II">
      <metainfo>
        <meta key="author" val="alex"/>
```

```
      </metainfo>
    </doc>
  </xobj>
</INSERT>
Instructions returned from DCMS to application
<!-- the handle of the inserted document is returned for later use  -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc002"/>
</RET>
```

Example 12: delete the document inserted in the example above

```
Instructions send from application to DCMS
<DELETE handle="doc002"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 13: use SYSTEM to save a docbase

```
Instructions send from application to DCMS
<SYSTEM>
  <flush path="c:\test.sep"/>
```

```
</SYSTEM>
<!-- instructions returned from DCMS to application -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 14: close the docbase (following example 1)

```
Instructions send from application to DCMS
<CLOSE  handle="docbase001"/>
instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

End of informative text.

(back to table)

GB-10:

The relationship between a Docbase and the root Docset is unclear. The term Docset is sometimes capitalised, sometimes not. If there is no distinction, the same form should be used throughout.

Added further wording to explain the relationship between Docbase and Docset, and attempted to ensure consistent spelling:

docbase: The root level of the UOML abstract document model. Abbreviated for “document base”, it is the container of one or many documents. A docbase contains one and only one root docset. [*Note*: The docbase is analogous to a file system on a modern operating system. The term docbase is derived from the term “database”. The docset is analogous to a directory within a file system on a modern operating system. The root docset is analogous to the root directory of a file system. *end note*].

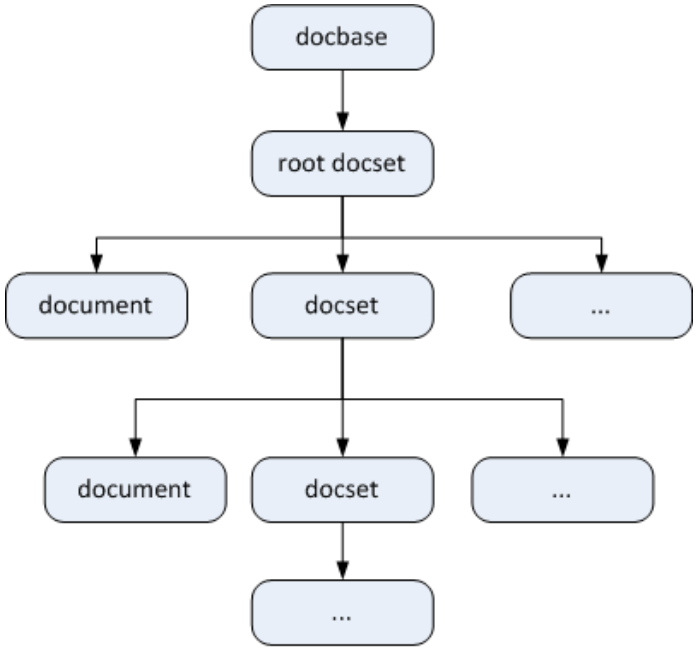
docset: A set of documents. A docset may contain one to many docsets. [*Note*: The docset is analogous to a directory within a file system on a modern operating system. *end note*].

Here is more explanatory wording in the Abstract Document Model Overview sub-clause

In the UOML abstract document model, documents are organized hierarchically via docbase, docset and document objects (see Figure 1). There are two sub-objects of a document object: document global objects and page related objects. Document global objects include font objects. Page related objects are organized hierarchically via pages, layers, object streams, command objects and graphics objects (see Figure 2).

One docbase shall have one and only one docset, known as the root docset. The root docset is the parent of all documents, similar to the root directory of a file system. As the container for documents, docsets may be nested (i.e., a docset may be a child of another docset). Figure 1 shows how a docbase, docset and document can construct a multiple level UOML-based tree structure, similar to a file system.

The following clauses provide a description of each object type.



1853 Figure1. UOML Abstract document Model 1

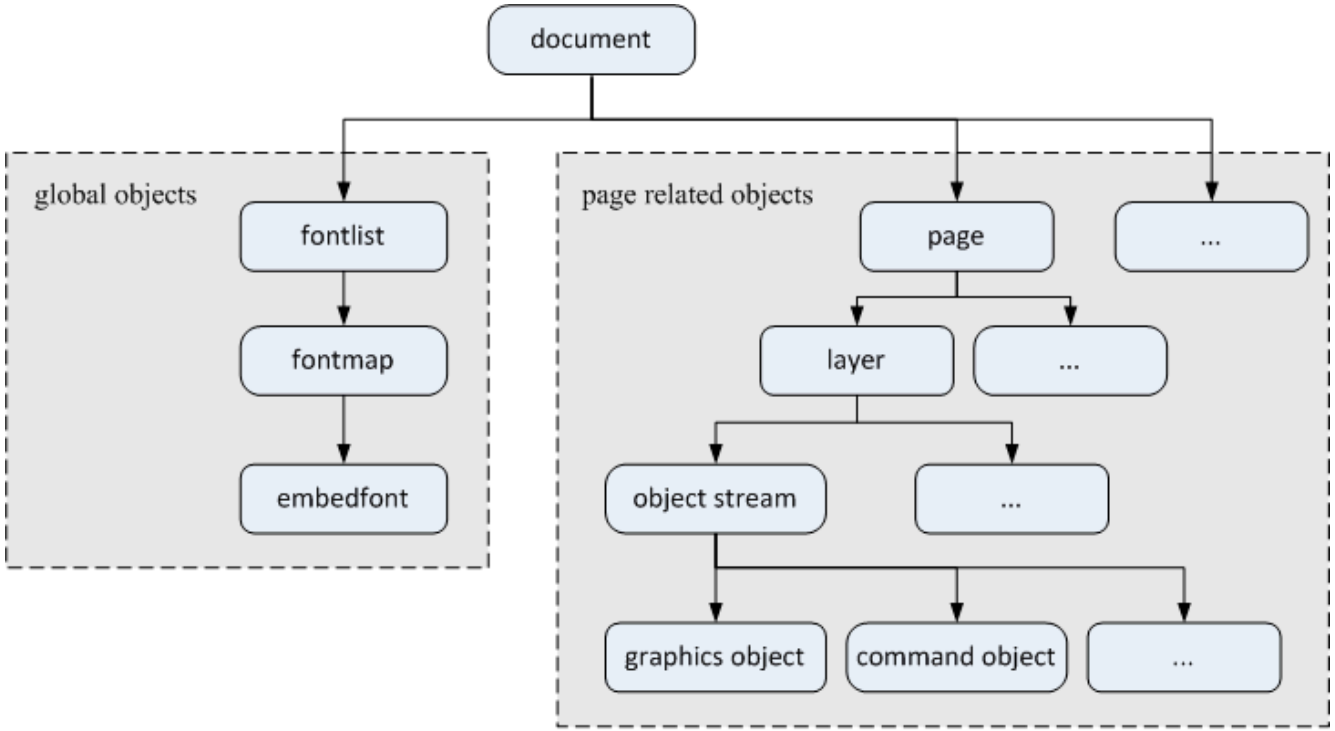


Figure 2. UOML Abstract Document Model 2

Docbase

The docbase is the root of the UOML abstract document model structure. A docbase has only one docset sub-object called the root docset [*Note*: Other docsets and documents are a docset's sub-objects. *end note*].

The root docset is generated automatically when the docbase is created (see Figure 1). In this specification, the docbase object is specified using DOCBASE (§4.3).

Sub-object: docset

Docset

A docset is an object whose sub-object can be a document, or another docset. In other words, a docset is a set of documents and/or docsets. In this specification, the docset object is specified using DOCSET (§4.4).

Sub-object: document, docset.

(back to table)

GB-11:

What is being defined here is both 'Page' and 'Layer'. Create separate definitions for Page and Layer.

Refined the definition of Layer in the Terminology clause.

layer: A page is composed of one or more layers. A layer has the same size as the page on which it is constructed. The visual appearance of a page is a combination of all of the layers of the page.

(back to table)

GB-12:
It is confusing to have a single definition for the term 'Path', but in the definition include the definition of another term 'path' that can easily be confused with the term being defined – particularly as the term is also used in capitalised form 'PATH'. The purpose of a Path is not clearly explained either in the definition here or in the body of the specification (Clause 2.5.10).

Changed part of the definition of Path to an informative note specifying the distinction between the two terms in question.

Path: A Path is a graphics object composed of straight and/or curved line segments, which may or may not be connected. [*Note*: that in this document, 'path' (all lowercase) refers to a filename, location of docbase or image file. This is different from this current definition of “Path” (with the uppercase ‘P’). *end note*]

Also modified some wording in the PATH element to clarify that it is the actual UOML element and not conceptual.

PATH

Semantics:

A Path specifies an open or closed region consisting of a collection of one or many subpaths, circles, ellipses, rectangles and round rectangles expressed using sub-elements. The PATH element itself does not contain any properties or data.

Properties: N/A

Sub-elements:

circle: CIRCLE type, defines a circle.

ellipse: ELLIPSE type, defines an ellipse.

rect: RECT type, defines a rectangle.

roundrect: ROUNDRECT type, defines a rectangle with round corners.

subpath: SUBPATH type, defines a subpath.

[*Example*: The following example demonstrates a PATH consisting of two sub elements: a rectangle and a circle.

```
        <INSERT pos="4">
          <xobj>
           <path>
            <circle center="167,251" radius="70" />
             <rect tl="124,135" br="345,257"/>
           </path>
          </xobj>
        </INSERT>
```

end example].

(back to table)

GB-13:
It is claimed that the specification uses XML syntax, but there are very few examples of how to express UOML in XML. The Explanatory Report states that the reference schema provided is non-normative. This presumably means that the few XML examples in the text of the specification are non-normative. See Example in Clause 2.6.1 and several Examples in Clause 3. Such Examples are helpful for understanding the specification and could usefully be provided in many other Clauses, but it should be made clear that these Examples are non-normative.

The provided UOML specification is indeed normative and a detailed example will be provided in the Appendix. All examples within the specification text, not in appendices, will be marked as non-normative. Here is an example:

[*Example*: Delete an object

```
<uoml:DELETE handle="img_obj_handle_xxx"/>
```

end example]

(back to table)

GB-16:
There is no normative reference to a specification for the image file formats 'BMP' or 'TIFF', yet these are referred to in normative text (in Clause 2.5.5).

Added related normative references in the Terminology clause.

[BMP] Bitmap Format. BMP. http://msdn.microsoft.com/en-us/library/at62haz6.aspx

[TIFF] ISO 12639:2004, *Graphic technology — Prepress digital data exchange — Tag image file format for image technology (TIFF/IT)*.

(back to table)

GB-17:
There is no normative reference to the definition of the "rgb" colour space referred to frequently in the text.

Added a normative reference to RGB.

[RGB] IEC 61966-2-1: 1999: Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB. International Electrotechnical Commission, 1999. ISBN 2-8318-4989-6 as amended by Amendment A1:2003.

(back to table)

GB-20:

The cardinalities of sub-elements and sub-objects are sometimes specified in narrative text but are not specified in the formal specifications of each object or element. In some cases the cardinality is completely missing.

Cardinalities are specified in the normative schema with “minOccurs” and “maxOccurs”. See example:

```
<xs:element name="RET">
	<xs:complexType>
		<xs:choice minOccurs="0" maxOccurs="unbounded">
			<xs:element name="intVal" type="uoml:INT"/>
			<xs:element name="floatVal" type="uoml:DOUBLE"/>
			<xs:element name="timeVal" type="uoml:TIME"/>
			<xs:element name="dateVal" type="uoml:DATE"/>
			<xs:element name="dateTimeVal" type="uoml:DATETIME"/>
			<xs:element name="durationVal" type="uoml:DURATION"/>
			<xs:element name="stringVal" type="uoml:STRING"/>
			<xs:element name="binaryVal" type="uoml:BINARY"/>
			<xs:element name="compoundVal" type="uoml:COMPOUND"/>
			<xs:element name="boolVal" type="uoml:BOOL"/>
			<xs:element name="longVal" type="uoml:LONG"/>
		</xs:choice>
	</xs:complexType>
</xs:element>
```

A UML diagram has been added in the clause describing the abstract document model showing cardinalities as well.

UML Diagram of UOML

The following is a UML diagram of the UOML abstract document model. It shows the tree structure of UOML along with cardinalities associated with the objects discussed in this clause.

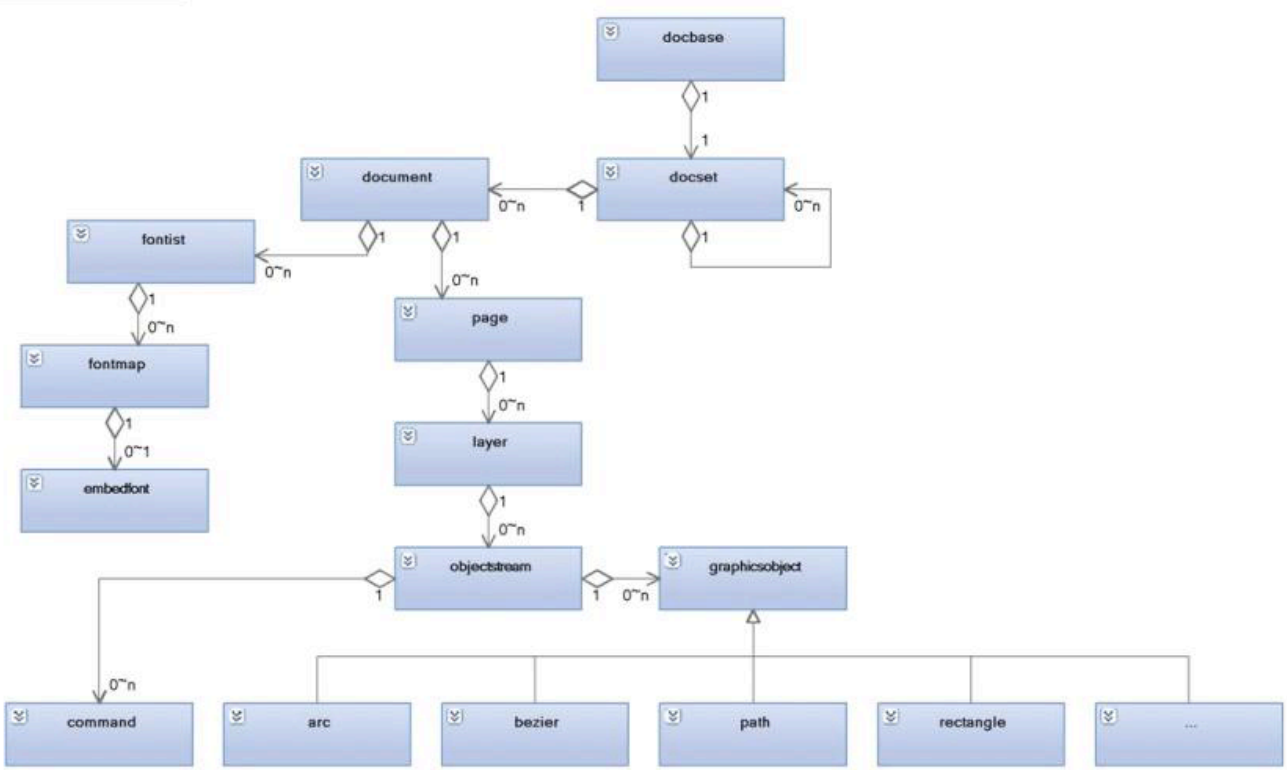


Figure 3. UML diagram of UOML abstract document model

General data ranges are also specified in the Data Ranges clause.

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.

4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

(back to table)

GB-22:

There are no definitions for the following terms that appear to have specific meanings in this specification:

- **sub-element**
- **direct sub-object**
- **indirect sub-object**

(Note that "sub-element" is not defined in XML 1.0.)

Added a definition for sub-element in Terminology. Also removed references to direct and indirect sub-object.

sub-element: In a UOML object XML representation, a sub-element is the child XML node of its parent XML node.

(back to table)

GB-24:
The sentence "Document Global Data can be used globalize among the document, it includes Metadata and Font" is poorly drafted and could therefore be subject to inconsistent interpretation. The term "Font" is used here and in several other places, but the terms "Font Data" and "Font Definition" are also used. It is unclear if these are synonymous. If they are, a consistent term should be used.

Restructured the document and repurposed the stated sentence to be more explanative and grammatically correct; removed instances of “Font Definition” and “Font Data”.

These sub-clauses are in the Abstract Document Model clause.

Overview

In the UOML abstract document model, documents are organized hierarchically via docbase, docset and document objects (see Figure 1). There are two sub-objects of a document object: document global objects and page related objects. Document global objects include includes font objects. Page related objects are organized hierarchically via pages, layers, object streams, command objects and graphics objects (see Figure 2).

Font

In the UOML abstract document model, three objects (fontlist, fontmap and embedfont), called font objects, are used to describe font information used in a document. A document object may contain zero or more fontlist sub-objects; a fontlist object may contain zero or more fontmap sub-objects; a fontmap may contain zero or one embedfont sub-object.

Fontlist is a list of fontmaps. Each fontmap describes one font used in the document, including font name and font sequential number used in the document. A document may optionally have font data embedded within it.

This sub-clause is in the UOML Objects clause.

Metadata

General information, such as the document's title, author, creation and modification date, is called metadata. Metadata is defined using keys and values. [*Note*: A key is not necessarily unique. A detailed specification of the keys and value falls outside the scope of this specification. *end note*]. In this specification, metadata is described using METALIST and META.

(back to table)

GB-26:
This Clause states that "Page Data may include 0 ... page(s)". Is it meaningful to have a document with no pages? Is it valid for a document to have no Global Data, i.e. no Metadata and no Font Data? It is not specified whether these can be omitted or are required in all documents.

Added clarifying wording in the Page sub-clause describing why 0 pages are permitted and that global objects are not required in a document.

Font

In the UOML abstract document model, three objects (fontlist, fontmap and embedfont), called font objects, are used to describe font information used in a document. A document object may contain zero or more fontlist sub-objects; a fontlist object may contain zero or more fontmap sub-objects; a fontmap may contain zero or one embedfont sub-object.

Fontlist is a list of fontmaps. Each fontmap describes one font used in the document, including font name and font sequential number used in the document. A document may optionally have font data embedded within it.

Page

A page object corresponds to a page in the document. Its sub-object is a layer object. A page object is composed of zero or more layer objects. The visual appearance of a page is a combination of all layers of the page.

Each page has its own size and resolution. The origin of a page's coordinate system is the top left corner of the page. The unit of a page's logical coordinate is defined by its resolution.

In this specification, the page object is described using PAGE (§4.7).

[*Note*: A document with no pages is permitted. It is an intermediate state. One can create such a document, then open and add pages at a future time. *end note*]

Sub-object: layer.

(back to table)

GB-27:
The property key is under-specified, particularly in terms of value constraints. For example, are two META objects allowed to have the same value of key?

Valid values will be described throughout the entire document. Also, wording has been added to state that the key/value properties need not be unique for a given instance of the element.

Metadata

General information, such as the document's title, author, creation and modification date, is called metadata. Metadata is defined using keys and values. [*Note*: A key is not necessarily unique. A detailed specification of the keys and value falls outside the scope of this specification. *end note*]. In this specification, metadata is described using METALIST and META.

METALIST

Semantics: A list of all the metadata in the document.

Properties: N/A

Sub-elements:

meta: META type.

META

Semantics: One item of metadata.

Properties:

key: character string value representing the key of metadata. [*Note*: A key is not necessarily unique. A detailed specification of the keys and value falls outside the scope of this specification. *end note*]

val: character string value representing the value of metadata.

Sub-elements: N/A

(back to table)

GB-28:

The sentence "EMBEDFONT, FONTMAP, FONTLIST in the order give above, the previous one is the sub-object of the latter one, and can be generated by UOML's INSERT instruction" is unclear and liable to a variety of interpretations.

The sentences pertaining to Font were rewritten for better clarity. In the Abstract Document Model clause.

In the UOML abstract document model, three objects (fontlist, fontmap and embedfont), called font objects, are used to describe font information used in a document. A document object may contain zero or more fontlist sub-objects; a fontlist object may contain zero or more fontmap sub-objects; a fontmap may contain zero or one embedfont sub-object.

Fontlist is a list of fontmaps. Each fontmap describes one font used in the document, including font name and font sequential number used in the document. A document may optionally have font data embedded within it.

In the UOML Objects clause.

Fontlist, fontmap and embedfont are called font objects. This clause gives the XML description of these objects.

(back to table)

GB-29:

The relationship between layers on the same page is unclear. A Layer apparently has no properties, but it possibly needs at least one property to specify whether a layer is opaque or transparent. It may be that all such properties are determined by the objects within the Layer, but at least some objects (e.g. Text) do not define opacity. There is no way in which the order of Layers on a page can be specified, other than implicitly in their order within an XML representation. There is therefore no certainty as to how to interpret the first step in Clause 2.4.4: "Repeat the following step[s] from the first layer to the last layer".

Added text in the Layer clause to explain the relationship between same page layers.

Layer

A layer object corresponds to one layer in a page. A layer is transparent. When a page has multiple layers, the order of each layer is the same as the order that it appears on the page. [*Note*: A page rendering engine should process layers in the aforementioned order. *end note*]

(back to table)

GB-30:

Graphics objects in other page description languages include many more properties than are listed for each of the objects specified in 2.5. It would appear that in UOML these properties are not specified on the graphics object itself, but are specified by a "command object". What is not clear is how to determine which command object influences which graphic object. The page processing model in 2.4.4 could be interpreted to imply either that a command object affects all graphics objects in the same OBJSTREAM, or that it affects only the graphics object that it follows (precedes?).

Added a Graphics State clause explaining how a command object affects the graphics state.

Graphics State

A DCMS shall maintain an internal data structure called the *graphics state* that holds the current graphics control parameters. The graphics state is initialized at the beginning of each layer with the default values specified in section §4.13. The rendering of a graphics object relies on the current parameters in the graphics state. A command object changes one or many parameters in the current graphics state.

(back to table)

GB-32:
The coordinate system being used to define Graphics Objects is inconsistently and incompletely defined. The terms "position", "point" and "coordinate" appear to be used interchangeably. No units are defined, nor any method to specify them.
The use of radians as the unit of measurement for angles is idiosyncratic and may cause interoperability problems with other page description techniques, such as PDF, XSL-FO and SVG, which use degrees.

Added clarifying wording in a Logical Coordinate System and Units clause to describe the logical coordinate system being used. Also discussion regarding resolution and radians were added.

Logical Coordinate System and Units

A UOML document uses a logical coordinate system. The terms *position*, *point* and *coordinate* may be used interchangeably. They refer to a logical point in the logical coordinate system. The origin of the logical coordinate system is the top left point. The direction of the x-axis is left to right. The direction of the y-axis is top to bottom.

The length of the units along each axis depends on the resolution property of the page. If the resolution of a page is x, the length of the unit along each axis is 1/x inch. A logical unit indicates one inch divided by the resolution of the page.

The resolution of each page is the same along the x and y axis.

UOML uses radians as the unit of measurement for angles. [*Note*: Though different from PDF, XSL-FO and SVG, conversion can be easily made without any loss of information. *end note*]

(back to table)

GB-33:
The relationship between an ellipse arc's clockwise and angle properties is unclear, making it appear that contradictory properties are possible (one "clockwise", the other "anti-clockwise").

Valid values for clockwise and angle will be provided, stating the boundaries between the two values.

ARC

Semantics:

An arc of an ellipse, specified by a starting, ending, and center position, along with a direction and angle.

Properties:

start: starting position of the arc.

end: ending position of the arc.

center: center of the arc's ellipse.

clockwise: the direction for arc is from the starting point to the ending point, which can be clockwise or counterclockwise. As a Boolean value, "true" represents clockwise and "false" represents counterclockwise.

angle: inclination from coordinate system's x-axis to arc's x-axis. It is specified using a radian value. A positive value represents counterclockwise and a negative value represents clockwise.

Sub-elements: N/A

(back to table)

GB-35:
Two radius properties are specified for the object ROUNDRECT. In other page descriptions languages rounded corners are generally circular, so only one radius is specified. If elliptical corners are being specified here, this should be made explicit. If a circular corner is required, it is unclear whether both properties xr and yr should be specified with the same value, or one of the properties should be omitted.

Added wording stating that the corners are elliptical in nature.

ROUNDRECT

Semantics:

A rectangle with round corners. The round corner of a round rectangle is a quarter of an ellipse.

(back to table)

GB-36:
It is unclear how a SUBPATH object would be expressed. An example would be helpful.

An example will be provided showing the expression of SUBPATH.

[*Example*: The following example demonstrates inserting of a Path object using INSERT instruction. The Path consists of two subpaths: a rectangle formed by four straight lines, and a curved line segment formed by Bezier curves.

```
<INSERT pos=”2” handle=”vs03”>
 <xobj>
  <path>
   <subpath data="s 214,193 l 368,193 l 368,298 l 214,298"/>
   <subpath data="s 417,206 B 417,186 426,167 435,167 B 443,167 452,230 452,293"/>
  </path>
 </xobj>
</INSERT>
```

end example].

(**back to table**)

GB-37:
It is unclear how a number of graphics objects are combined together to form a PATH. This needs further explanation with illustrative examples.

An example will be provided showing the expression of PATH.

[*Example*: The following example demonstrates a PATH consisting of two sub elements: a rectangle and a circle.

```
<INSERT pos="4">
  <xobj>
   <path>
    <circle center="167,251" radius="70" />
     <rect tl="124,135" br="345,257"/>
   </path>
  </xobj>
</INSERT>
```

end example].

(back to table)

GB-38:
It is unclear how the spaces property is to be interpreted. How is the "origin" of a character defined? How is distance specified? Can distances be negative? Does the number of comma-separated values have to be the same as the number of characters in the string? Do these values override the widths of the characters as specified by the font used? How are these values affected by the action of an applicable command object (which may specify font, size, weight, other transformations)?

Added further clarification on the origin and the spaces properties in the TEXT object.

TEXT

Semantics:

Text, specified using an origin, encoding information, text data and an optional character spacing list.

Properties:

origin: the coordinate of the first character's origin. The origin of a character is defined by its font information.

encode: character set or encoding of text data. The valid value for this property should be one of the character encodings registered (as charsets) with the Internet Assigned Numbers Authority [IANA-CHARSETS], otherwise it should use names starting with an x- prefix.

text: character data contained in text, base64 encoded string data.

spaces: an optional, ordered set of distances that specifies distances between adjacent characters' origins, separated by a comma.

The origin of a character refers to the point (0, 0) in the coordinate system of the character glyph, as illustrated in the Figure 4. When a text object with only one character is specified and the text object has coordinate (x, y), the rendering engine should place the origin of the character at (x, y) and render the character.

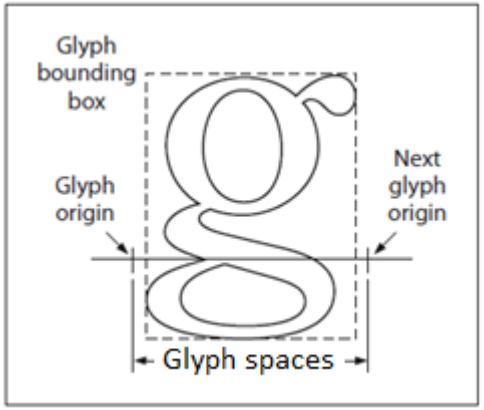


Figure 4. spaces of text

The spaces property is the offset or distance between the x coordinates of two adjacent characters. It is always positive. The number of comma-separated values shall be one fewer than the number of characters in the string. The values should override the widths of the characters as specified by the font used. The values are used to calculate the coordinate to place the origin of each character.

Sub-elements: N/A

(**back to table**)

GB-39:
A form of BNF notation is used, but there is no normative reference to the specification of this notation.
The first sentence includes the word 'path' with no capital letter, but 'Path' is probably intended here.

Added a normative reference to BNF and changed ‘path’ to ‘Path’.

ISO/IEC 14977:1966, *Information technology — Syntactic metalanguage — Extended BNF*.

(back to table)

GB-41:
The meaning of 'red', 'green', 'blue' and 'alpha' is not clear, because the colour standard or reference specification is not stated.

Added a normative reference to RGB.

[RGB] IEC 61966-2-1: 1999: Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB. International Electrotechnical Commission, 1999. ISBN 2-8318-4989-6 as amended by Amendment A1:2003.

(back to table)

GB-46:
It is unclear how this Clause relates to Clause 2.6.2.13. How is the "direction" of a character defined?

Clarified the meaning of character direction.

CHAR_DIR

Semantics: Set the current character direction (e.g., the direction in which a character is rendered). The heading direction is from the bottom of a character to the top.

Properties:

v1: a character string representing the character direction. The possible values for this property are HEAD_LEFT, HEAD_RIGHT, HEAD_TOP and HEAD_BOTTOM. HEAD_LEFT is the character's heading direction is left. HEAD_RIGHT is the character's heading direction is right. HEAD_TOP is the character's heading direction is up. HEAD_BOTTOM is the character's heading direction is down.

Sub-elements: N/A

(back to table)

GB-47:
The property v1 of the command object CHAR_SLANT is inadequately specified. It is not clear whether "right" always means the same thing, regardless of reading direction, or whether it actually means "in the direction of reading".

Clarified that slants are regardless of reading direction.

CHAR_SLANT

Semantics: Set the slant of the current character.

Properties:

v1: a floating point number, representing the character slanting radian, regardless of reading direction. 0～π/2 represents right slant, 3π/2～2π represents left slant, and 0 represents non-slant; other values are not used.

Sub-elements: N/A

(back to table)

GB-48:
It is unclear how to interpret a decimal value between 0 and 1 as a character weight.

The algorithm will be specified in the CHAR_WEIGHT clause.

CHAR_WEIGHT

Semantics: Set the current character weight. The default value is 0. The thickness of a character stoke shall be the normal thickness plus weight*(character height). The minimum thickness of a character's stroke is zero.

Properties:

v1: a floating point number, ranging between -1 to 1, inclusively, representing the character weight.

Sub-elements: N/A

(back to table)

GB-49:
How are the properties of CHAR_STYLE to be interpreted? Is this implementation-dependent? If so, this needs to be stated. If not, the allowed values of this property need to be enumerated and defined.

Specified normatively the details on the three properties of CHAR_STYLE. A detailed algorithm has been added to the specification.

CHAR_STYLE

Semantics: Set the current character style.

Properties:

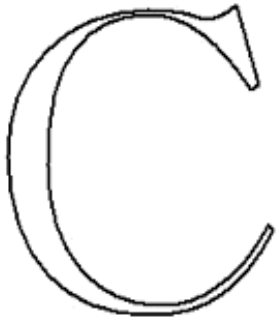
v1: a character string, representing the character style. The possible values for this property are SHADOW, HOLLOW and OUTLINE, or some combination of the three, separated by commas. If the string is set to empty, then any previous setting is cleared.

SHADOW: set shadow style. If this character style is set, then the following algorithm is used to render the shadow effect:

- If SHADOW_NEG (§4.11.2.30) is false, the character is extended with a distance of SHADOW_LEN (§4.11.2.27) along the shadow direction (§4.11.2.28), then a hollowed character with raster operation ROP_COPY is drawn in the original position. The border width of the hollowed character is SHADOW_WIDTH (§4.11.2.26).

- If SHADOW_NEG is true, the character position is moved with a distance of SHADOW_LEN along the shadow direction, and extended SHADOW_WIDTH along the shadow direction; then the character is drawn in the original position with background color and raster operation ROP_COPY, and extended with a distance SHADOW_LEN along the shadow direction; then in the original position, a character with normal color and raster operation ROP_COPY is drawn.

HOLLOW: set hollow style. If this character style is set, a line with thickness HOLLOW_BORDER (§4.11.2.35) should be drawn along the outline of the character.



OUTLINE: set outline style. If this character style is set, a line with thickness OUTLINE_BORDER (§4.11.2.33), and with distance OUTLINE_WIDTH (§4.11.2.34) from the outline of the character, should be drawn along the outline of the character.

Sub-elements: N/A

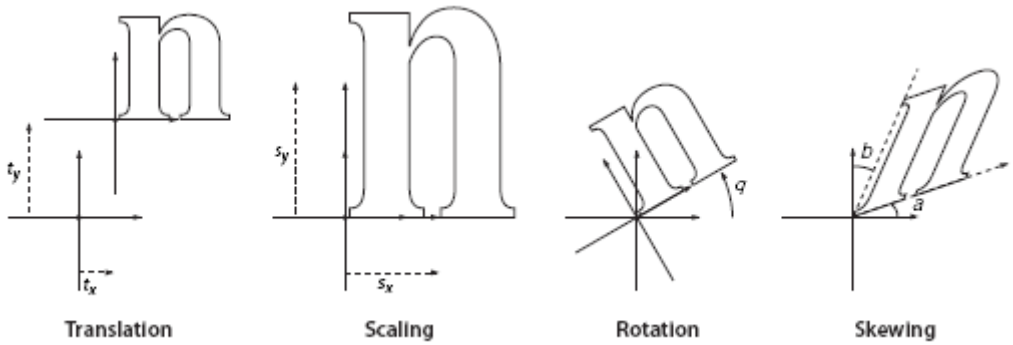
(back to table)

GB-50:
It is unclear whether this command applies to each character individually within a TEXT object, or to the TEXT object as a whole.

Described that the command applies to each character individually within a TEXT object.

TEXT_MATRIX

Semantics: Set the current text transformation matrix. This command applies to each character individually within a TEXT object. The visual effect of transforming a character is shown below:



Properties: N/A

Sub-elements:

matrix: element of the MATRIX (§4.11.3.2) type, responsible for transforming coordinates of text.

(back to table)

GB-51:
The command objects PUSH_GS and POP_GS make reference to a "stack", but no stack is defined as part of the page processing model in Clause 2.4.4.

Defined the term ‘graphics state stack’ in the Terminology clause.

graphics state stack: A sequence of graphics states where the first one in is the last one out. A DCMS shall maintain a stack for graphics states, called the graphics state stack**.** [*Note*: The command object PUSH_GS saves a copy of the current graphics state onto the stack. The command object POP_GS restores the saved copy, remove it from the stack and make it the current graphics state. *end note*]

(back to table)

GB-52:

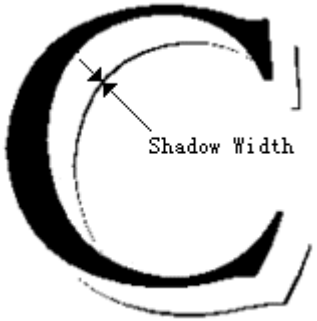
It is unclear how to interpret the way in which these command objects specify a shadow to be applied to a character. The object SHADOW_WIDTH appears to refer to the thickness of the outline of a shadow. If so, this should be clarified. It would help if the name were clearer. The object SHADOW_LEN appears to refer to the displacement of the shadow with reference to the character, and presumably the direction of displacement is given by the object SHADOW_DIR. If so, this should be clarified. The range of values specified for SHADOW_DIR look somewhat restrictive compared with what is possible in applying shadow to text in many graphics design systems. Why is the direction not specified as an angle (in radians, of course)? What is meant by adjusting the coordinates of a character when the direction of shadow is to the left and top? Should there not be equivalent command objects for adjusting the coordinates of a character when the shadow is oriented in some other direction?

If the command object SHADOW_NEG is applied, what determines the thickness of the border of the character? Is this SHADOW_WIDTH?

Added further description to figures and clarification to SHADOW_WIDTH, SHADOW_LEN, SHADOW_NEG, SHADOW_DIR. The four directions specifying the properties for SHADOW_DIR, though simplified and restrictive, are able to meet shadowing demands in most cases. Also, this is much easier to implement compared to the being able to specify a direction at an arbitrary angle.

SHADOW_WIDTH

Semantics: Set the border width of the current character shadow. SHADOW_WIDTH represents the thickness of the outline of a shadow.



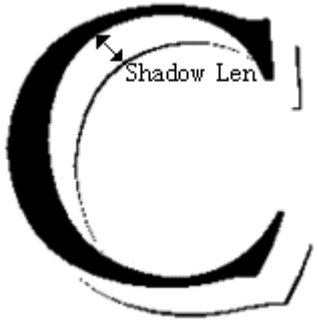
Properties:

v1: a non-negative floating point number, representing the shadow border width.

Sub-elements: N/A

SHADOW_LEN

Semantics: Set the length of the current character shadow. SHADOW_LEN represents the distance of the shadow from the corresponding point of the character along the character shadow direction (§4.11.2.28) to the edge of the shadow width.



Properties:

v1: a non-negative floating point number, representing the character shadow length.

Sub-elements: N/A

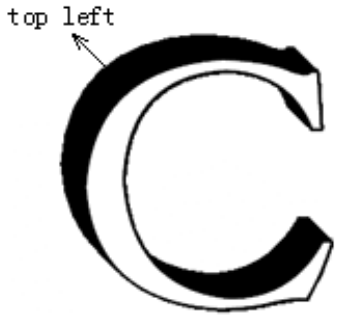
SHADOW_DIR

Semantics: Set the direction of the current character shadow

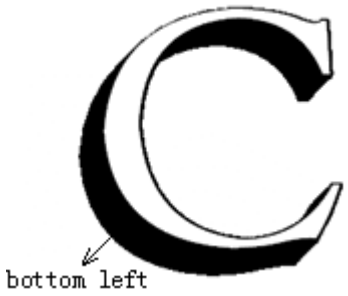
Properties:

v1: a character string. The possible values for this property are SHADOW_LT, SHADOW_LB, SHADOW_RT and SHADOW_RB.

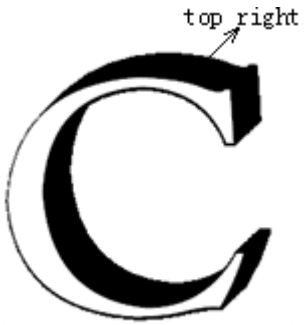
SHADOW_LT: the character shadow direction is top left.



SHADOW_LB: the character shadow direction is bottom left.



SHADOW_RT: the character shadow direction is top right.



SHADOW_RB: the character shadow direction is bottom right.

Sub-elements: N/A

SHADOW_ATL

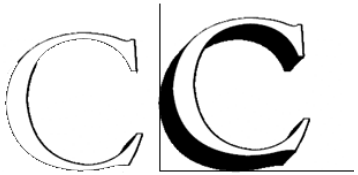
Semantics: Set whether to adjust the coordinates of a character when the direction of character shadow is to the left or bottom.

Properties:

v1: a Boolean value, representing whether to alter the coordinates of a character. The value ‘true’ specifies that the coordinates are altered.

Sub-elements: N/A

[*Example*: Illustrated in the figures below, when a character is shadowed, the bounding box of its outline is bigger. If two characters that are not shadowed are adjacent, their baselines are aligned horizontally. A shadow effect will break this horizontal alignment. Also, a shadow to the left will occupy the space between this character and its left neighbor. When a rendering engine draws the character, it can position the character based on the specific coordinate; or it can adjust the coordinate so that the bottom left point of the shadowed character’s outline bounding box moves to the specific coordinate. This is made by offset x or y coordinates by the distance of SHADOW_LEN divided by the square root of 2. When the shadow is to the bottom of the character, subtract y by the distance; when the shadow is to the left, add x by the distance. Make both adjustments when the shadow is to the bottom left. This explains the parameter SHADOW_ATL. When SHADOW_ATL is false, the specific coordinate is used without adjustment; when it is true, an adjustment should be made. The first figure illustrates the effect before adjustment, while the second figure illustrates the effect after adjustment.



end example]

SHADOW_NEG

Semantics: Set the current shadow character as an intaglio character as illustrated in the following figures.

SHADOW_NEG is false

SHADOW_NEG is true

Properties:

v1: a boolean value, representing whether the current shadow character is an intaglio character. A ‘true’ value specifies an intaglio character.

Sub-elements: N/A

(back to table)

GB-53:
It is unclear how a CLIP_AREA command object is to be applied to the OBJSTREAM that contains it. Is the Path specified by this object relative to the Page or relative to some coordinate origin for the OBJSTREAM?

Clarified how a CLIP_AREA is applied to an OBJSTREAM in the CLIP_AREA clause.

CLIP_AREA

Semantics: Set the current clip area

Properties: N/A

Sub-elements:

cliparea: PATH type, representing the new clip area.

The Path specified by a CLIP_AREA command object is relative to the page. The portions of graphic objects that lie outside of the current clip area are not rendered.

(back to table)

GB-56:

The property value GET_PAGE_BMP appears to return an object that is not defined by this specification. If a page bitmap is simply a page containing a single graphics object in BMP format, this should be explained.

Defined page bitmap in the Terminology clause.

page bitmap: A raster image that represents the visual appearance of the page. The number of pixels of the raster image depends on the resolution of the raster image. The number of pixels in the horizontal direction equals the page width multiplied by the resolution; the number of pixels in the vertical direction equals the page height multiplied by the resolution. [*Note*: The resolution is the same for both the horizontal and vertical direction. *end note*]

(back to table)

GB-58:
While 1.3 contains a reference to the XML Schema Part 2: Datatypes Second Edition, there is no statement that the data type referred to in this Clause (or any other) is to be taken from that W3C specification, or of which sub-formats of XML dates are valid. If one were to specify a date such as 0101 to indicate first of January, without specifying the year, is this valid?

While the data type of DATE is specified normatively in the UOML XML Schema, valid values for DATE will be described. A normative reference has been added to ISO 8601 and a Data Ranges clause has been added.

[DATE] ISO 8601:2004**,** *Data elements and interchange formats – Information Interchange – Representation of dates and times.*

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.
4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

(back to table)

JP-01:
It is not clear what the scope of this standard is and what is intended to be standardized. As demonstrated by the sheer number of undefined terms as well as the poor quality of BNFs, schemas, and API descriptions, the quality of this standard is extremely low.

Added a Scope clause to the specification and have improved the overall specification for greater clarity.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

(back to table)

JP-08:
The BNF notation should be described in accordance with ISO/IEC 14977.

Added a normative reference to ISO/IEC 14977 in the Normative Reference clause.

ISO/IEC 14977:1966, *Information technology — Syntactic metalanguage — Extended BNF*.

(back to table)

KR-01:
There is not provide the explanation of the exact mark scheme(URL, URI etc.) about path attribute-value of OPEN elements. It gives the definition of Return value, but it isn't also exist the definition of HANDLE which explain the return value. Also, No logic and scheme is presented as RET and it may be difficult to use for real user of this standards due to lack of information.

Provided further explanation regarding the path property of OPEN. HANDLE and RET are defined in the normative UOML XML schema provided with the specification and also now added as an informative Appendix to the UOML specification.

OPEN

Semantics:

OPEN creates or opens a docbase.

Properties:

create: a Boolean value representing whether to create a docbase if it does not exist. Specifying 'true' will create the docbase. The default value is 'true'.

del_exist: a Boolean value, representing whether to delete the docbase if it already exists. Specifying 'true' deletes the existing docbase. The default value is 'false'.

path: a character string value, representing the location of a docbase. There is no defined format for the path value (e.g., URI, URL, fully-qualified file system directory path, absolute value, relative value, etc.). Valid values for this property, and their appropriate interpretation, are implementation-defined. [*Note*: A path should be a format such that it could be used to find the location of the docbase. *end note*]

Sub-elements: N/A

Return value:

If OPEN succeeds, the returned RET element contains a 'stringVal' sub-element with the 'name' property as the handle and the 'val' property represents the handle of the docbase. [*Note*: The syntax of the handle value is implementation-defined and has no relationship to other handles returned by the given DCMS nor to other handles returned by another DCMS, even for the creation of the same document. *end note*]

If OPEN fails, the return value is defined by RET (§3.9).

[*Example*:

Create a docbase, named 1.sep. If the DCMS successfully processed the OPEN instruction, it will return a RET instruction.

```
<OPEN path="/home/admin/storage/1.sep" create="true" del_exist="false"/>

Return element if OPEN succeeds:

<RET>
 <boolVal name="SUCCCESS" val="true"/>
 <stringVal name="HANDLE" val="db_handle_xxxxx"/>
</RET>

Return element if OPEN fails:

<RET>
 <boolVal name="SUCCCESS" val="false"/>
 <stringVal name="ERR_INFO" val="required resource not available"/>
</RET>
```

end example]

RET is also defined as a separate instruction within the UOML specification as well.

RET

Semantics:

RET is the return value from the DCMS to the application software. RET may contain one or more return values, and each return value is represented by one sub-element (e.g., boolVal, stringVal, intVal, floatVal, compountVal, etc.).

The 'name' property of the sub-element represents the name of the return value.

If the return value is a simple type, the 'val' property of sub-element contains the return value.

If the return value is a compound type, a sub-element will be added under the corresponding sub-element to represent the compound return value.

RET contains at least one 'boolVal' sub-element to describe whether the operation was successful or not. Its 'name' property is SUCCESS, and its 'val' property is either ‘true’ or ‘false’, depending on the success of the operation.

When the operation fails, RET also contains one 'stringVal' sub-element. Its 'name' property is ERR_INFO, and its 'val' property describes the failure information, in an implementation-defined way. [*Note*: For other return values, check the definition of the concrete UOML instruction for reference. *end note*]

[*Example*: `<boolVal name="SUCCESS" val="true"/>` *end example*]

Properties: N/A

Sub-element:

intVal: integer type return value, INT type

floatVal: float type return value, DOUBLE type.

TimeVal: time type return value, TIME type.

DateVal: date type return value, DATE type.

DateTimeVal: date and time type return value, DATETIME type.

DurationVal: time duration type return value, DURATION type.

StringVal: string type return value, STRING type.

BinaryVal: binary type return value, BINARY type.

CompoundVal: compound type return value, COMPOUND type.

BoolVal: boolean type return value, BOOLEAN type.

[*Example*:

Return two values.

```
  <RET>
    <boolVal name="SUCCESS" val="false"/>
    <stringVal name="ERR_INFO" val="required resource not available"/>
  </RET>
```

end example]

(back to table)

KR-03:
There is no exist the exactly definition of current object.

Added an informative note describing object and current object.

USE

Semantics:

USE sets an object as the current object. [*Note*: USE sets an object in the document to the current object of focus. The current object is used when the destination object is not specified within an instruction (e.g. INSERT). *end note*]

(back to table)

KR-04:
It gives the definition of operation method using the GET_SUB, GET_SUB_COUNT, GET_PROP, GET_PAGE_BMP value, but there is not explained the detailed sub-elements and is not clear the unit of attribute-value used in the sub elements.

All sub-elements are defined in the normative XML Schema accompanying the UOML specification. Unit information has also been added to the GET_SUB, GET_SUB_COUNT, GET_PROP, GET_PAGE_BMP within the revised specification.

GET

Semantics:

GET retrieves information such as a sub-object handle, the count of sub-objects, the property value of an object, or a page bitmap.

Properties:

usage: a character string value, representing the usage of GET. The possible values of this property are GET_SUB, GET_SUB_COUNT, GET_PROP, GET_PAGE_BMP, representing getting a sub-object, getting the sub-object count, getting properties, and getting a page bitmap, respectively.

handle: a character string value, representing the object handle of the current operation. This property is optional. If this property is not used, then the current handle set by the USE instruction is used.

Sub-elements:

pos: used when usage=GET_SUB.

Property of this sub-element:

val: specifies the position number of the specified sub-object, starting from 0.

Sub-element of this sub-element: N/A

property: used when usage=GET_PROP.

Property of this sub-element:

name: specifies the name of the property whose value is returned, if *name* is an empty string, the type of the object is retrieved.

Sub-element of this sub-element: N/A

disp_conf: used when usage=GET_PAGE_BMP.

Properties of this sub-element:

end_layer: specifies the handle of the end layer of the operation (the drawing operation ends at this layer and this layer is not drawn any more)

resolution: represents resolution of bitmap

format: represents the bitmap format. The only valid value is "bmp", representing the uncompressed BMP format.

output: represents whether to put out to the file or to the memory. Possible values for this property are FILE or MEMORY;

addr: represents the path of output file or memory address.

Sub-element of this sub-element:

clip: represents clip area for output, PATH type.

Usage value / Return value:

The return value is based on the usage value:

- GET_SUB_COUNT: If the usage is GET_SUB_COUNT, this indicates to get the number of sub-objects of this specific object. In this case, there is no sub-element needed for the GET instruction. The return value, which is returned via the RET instruction, contains one 'intVal' sub-element. Its 'name' property is "sub_count" and the 'val' property represents number of sub-objects.

[*Example*:

Get the total number of sub-objects of the specific object:

```
<GET handle="obj_handle_xxx" usage="GET_SUB_COUNT"/>
```

RET instruction returns the number:

```
<RET >
  <boolVal name="SUCCESS" val="true"/>
```

```
        <intVal name="sub_count" val="1"/>
      </RET>
```

end example]

- GET_SUB: If the usage is GET_SUB, this indicates to get the handle of some specific sub-object. In this case, GET shall contain the sub-element of 'pos'. The return value, which is returned via the RET instruction, contains one 'stringVal' sub-element. Its 'name' property is “handle” and its 'val' property represents the sub-object's handle.

[*Example*:

Get a specific sub-object handle:

```
<GET handle="obj_handle_page01" usage="GET_SUB">
   <pos val="0"/>
</GET>
```

RET instruction returns the handle of the sub-object:

```
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="obj_handle_layer01"/>
</RET>
```

end example]

- GET_PROP: If the usage is GET_PROP, this indicates to get some specific property of a specific object. If the name property is a non-empty string, GET shall contain the sub-element of 'property'. If the operation succeeds, the sub-element of return value, which is returned via RET instruction, is variant; the sub-element name relies on the type it has retrieved, the 'name' property of the sub-element is the property name to get, 'val' property is the value of the property; otherwise if the name property is an empty string, the RET instruction returns a stringVal value representing the type of the object, which is the element name of the XML description of the object without the namespace prefix.

[*Example*:

Get specific property of the object

```
<GET handle="obj_handle_xxxxx" usage="GET_PROP">
   <property name="start"/>
```

```
</GET>
```

RET instruction returns the start property, which is a coordinate:

```
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="start" val="200,300"/>
</RET>
```

end example]

- GET_PAGE_BMP: If the usage is GET_PAGE_BMP, this indicates to get the specific page bitmap. In this case, GET shall contain the sub-element 'disp_conf'. The requested bitmap should be placed/returned where the ‘addr’ and ‘output’ property of the ‘disp_conf’ element is specified.

[*Example*:

Get specific page's bitmap

```
<GET handle="page_obj_handle_xxx" usage="GET_PAGE_BMP">
   <disp_conf format="bmp" output="FILE" end_layer="1" resolution="600"
       path="/home/admin/output/page.bmp">
      <clip>
         <subpath data="s 0,0 l 3000,0 l 3000, 5000 l 0, 5000 l 0,0"/>
      </clip>
   </disp_conf>
</GET>
```

end example]

- When GET fails, the return value is defined by RET.

[*Example*:

```
<RET>
   <boolVal name="SUCCESS" val="false"/>
   <stringVal name="ERR_INFO" val="disk full"/>
</RET>
```

end example]

Also a Data Ranges sub-clause has been added.

Data Ranges

The following are the general rules for data ranges:

1. Unless otherwise specified, all numeric values may be positive, negative or zero.
2. Positive, negative, or zero values are allowed for coordinates and points in the logical coordinate system (e.g. -1, 3).
3. Integer values use a 32-bit memory representation; the range of integer values is from -2,146,483,648 to 2,147,483,647.
4. IEEE single-precision floating numbers are used; the valid range is -3.403e38 to 3.403e38.
5. API calls that set values outside a valid range (either specifically specified or within the ranges above) will fail with a return of RET.
6. A special case is COLOR_RGB. RGB32 is used, thus each property of COLOR_RGB(r, g, b, a) falls within a range of 0-255.
7. Valid ranges and formats for a date are specified in ISO 8601.

(back to table)

NL-01:
The scope of the standard is not clear.

Added a Scope clause to the specification.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

(back to table)

NL-03:
Because of the lack of detail the standard can't be implemented.

Expanded the Conformance clause, provided more information and examples in order to aid in possible implementations.

The text in this OASIS standard is divided into *normative* and *informative* categories. Unless documented otherwise, all features specified in normative text of this OASIS standard shall be implemented. Text marked informative (using the mechanisms described in §1.5) is for information purposes only. Unless stated otherwise, all text is normative.

Use of the word "shall" indicates required behavior.

Any behavior that is not explicitly specified by this OASIS standard is implicitly unspecified (§4).

DCMS Conformance

A UOML Document Management System (DCMS) has conformance if it implements all of the UOML instructions in compliance with the syntax and semantics in this OASIS standard.

Application Conformance

A UOML application is conformant if both of the following are true:

The application issues UOML instructions to the DCMA as specified in this OASIS standard; and
The application parses the return instructions from the DCMS according to this OASIS standard.

Added an Annex of Detailed Examples.

This annex is informative.

The examples below demonstrate the usage of many of the UOML instructions. Each example is followed by a corresponding "RET "instruction.

The XML string of a UOML instruction may be preceded by a prolog to specify the character encoding of the XML string. If default encoding is UTF-8, the prolog, `<?xml version="1.0" encoding="UTF-8"?>`, may be omitted. The default namespace for the XML string is:
`urn:oasis:names:tc:uoml:xmlns:uoml:1.0`.

Example 1: open a docbase

Instructions sent from application to DCMS

```
<OPEN create="false" del_exist="false" path="c:\test.sep"/>
```

Instructions returned from DCMS to application

```
<!-- the string value "docbase001" is the opened docbase's handle for later use -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docbase001"/>
</RET>
```

Example 2 : get the root docset of the docbase (following example 1)

Instructions sent from application to DCMS

```
<!-- since each docbase has one and only one sub-object, to get the root docset is just to get the first sub-object of docbase whose handle is returned by example 1 -->
<GET handle="docbase001" usage="GET_SUB">
  <pos val="0"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="docset001"/>
</RET>
```

Example 3: get the number of sub-objects of the root docset (following example 2)

Instructions sent from application to DCMS

```
<GET handle="docset001" usage="GET_SUB_COUNT"/>
```

Instructions returned from DCMS to application

```
<!-- the return value of 3 indicates the root docset has 3 sub-objects -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <intVal name="sub_count" val="3"/>
</RET>
```

Example 4: get the third sub-object of the docset (following example 3)

Instructions sent from application to DCMS

```
<GET handle="docset001" usage="GET_SUB">
  <pos val="2"/>
</GET>
```

Instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc001"/>
</RET>
```

Examples 5: get the type of a object using the empty string as the name of the property (following example 4)

Instructions sent from application to DCMS

```
<GET  usage="GET_PROP" handle= "doc001 ">
```

```
  <property name=""/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="" val="DOC"/>
</RET>
```

Example 6: get the metadata of the document (following example 4)

```
Instructions sent from application to DCMS
<GET  usage="GET_PROP" handle= "doc001 ">
  <property name="metainfo"/>
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <compoundVal name="metainfo">
       <metalist>
             <meta key= "title" val="UOML Part I"/>
             <meta key="author" val="UOML TC"/>
       </metalist>
  </compoundVal>
</RET>
```

Example 7: get page bitmap of a page

Instructions sent from application to DCMS

```
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
<GET usage="GET_PAGE_BMP" handle="page001">
  <disp_conf addr="c:\test.bmp" end_layer="8" format="bmp" output="FILE"
resolution="640">
    <clip>
      <ellipse angle="45" center="10,20" xr="30" yr="40"/>
      <roundrect br="70,80" tl="50,60" xr="90" yr="100"/>
      <subpath data="s 214,193 1 368,193 1 368,298 1 214,298"/>
    </clip>
  </disp_conf>
</GET>
```

Instructions returned from DCMS to application

```
<!-- the bmp format of page bitmap data has been saved in the file c:\test.bmp as requested -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 8 : get first layer of a page

Instructions sent from application to DCMS

```
<!-- the page object's handle is supposed to have already obtained of value "page001" in prior instructions(using GET) -->
```

```
<!-- since page has only layer objects as its sub-objects, get sub-objects is the same to get layer objects -->
<GET  handle="page001" usage="GET_SUB">
  <pos val="0"/>
</GET>
Instructions returned from DCMS to application
<RET >
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="layer001"/>
</RET>
```

Example 9: set a text object as the current object

```
Instructions send from application to DCMS
<!-- the text object's handle is supposed to have already obtained of value "text001" in prior instructions(using GET) -->
<USE  handle="text001"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Examples 10: get spaces property of a text object (following example 9)

```
Instructions send from application to DCMS
<GET  usage="GET_PROP">
  <property name="spaces"/>
```

```
</GET>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="spaces" val="50,55"/>
</RET>
```

Example 11: insert a document into a docset (following example 2)

```
Instructions send from application to DCMS
<INSERT handle="docset001">
  <xobj>
    <doc name="UOML part II">
      <metainfo>
        <meta key="author" val="alex"/>
      </metainfo>
    </doc>
  </xobj>
</INSERT>
Instructions returned from DCMS to application
<!-- the handle of the inserted document is returned for later use  -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
  <stringVal name="handle" val="doc002"/>
```

```
</RET>
```

Example 12: delete the document inserted in the example above

```
Instructions send from application to DCMS
<DELETE handle="doc002"/>
Instructions returned from DCMS to application
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 13: use SYSTEM to save a docbase

```
Instructions send from application to DCMS
<SYSTEM>
  <flush path="c:\test.sep"/>
</SYSTEM>
<!-- instructions returned from DCMS to application -->
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

Example 14: close the docbase (following example 1)

```
Instructions send from application to DCMS
```

```
<CLOSE  handle="docbase001"/>
```

instructions returned from DCMS to application

```
<RET>
  <boolVal name="SUCCESS" val="true"/>
</RET>
```

End of informative text.

(back to table)

NO-01:
We do not feel that this specification is of the quality required for it to become an International Standard. Further, we feel that it is not sufficiently clear what problem the specification exists to solve. Therefore, if SC34 is to perform further work on this specification, the first step should be to clarify the background and requirements for this proposed standard.

Added new clauses to the specification (e.g., Scope) and modified other clauses for better clarity.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

Notational Conventions

The following typographical conventions are used in this OASIS standard:

1. The first occurrence of a new term is written in italics, as in "*normative*".
2. In each definition of a term in §1.1 (Terminology), the term is written in bold, as in "**behavior**".

Acronyms and Abbreviations

This clause is informative

The following acronyms and abbreviations are used throughout this OASIS standard:

DCMS — Docbase Management System

IEC — the International Electrotechnical Commission

ISO — the International Organization for Standardization

UOML — Unstructured Operation Markup Language

W3C — World Wide Web Consortium

End of informative text

General Description

This OASIS standard is divided into the following subdivisions:

1. Front matter (clause 1);
2. Main body (clauses 2-4);
3. Conformance (clause 5);
4. Annexes

Examples are provided to illustrate possible forms of the constructions described. References are used to refer to related clauses. Notes may be provided to give advice or guidance to implementers or programmers.

The following form the normative pieces of this OASIS standard:

- Clauses 1 (except subclauses 1.4, 1.6, and 1.8) and 2–5

The following form the informative pieces of this OASIS standard:

- Introductory text in clause 1
- Subclauses 1.4, 1.6, and 1.8
- All annexes
- All notes and examples

Except for whole clauses or annexes that are identified as being informative, informative text that is contained within normative text is indicated in the following ways:

1. [*Example:* code fragment, possibly with some narrative … *end example*]
2. [*Note:* narrative … *end note*]
3. [*Rationale:* narrative … *end rationale*]
4. [*Guidance*: narrative … *end guidance*]

Overview

This clause is informative

This OASIS standard specifies an instruction set of XML elements and attributes describing operations on unstructured, fixed-layout documents. These instructions are for the processing of these documents to accomplish various functionality, such as display and edit.

UOML is to unstructured documents as SQL (Structured Query Language) is to structured data. UOML is expressed using standard XML via an instance of an XML schema. UOML handles fixed-layout documents and its associated information (e.g., metadata, security rights, etc.) Fixed-layout- documents are two-dimensional and contain static paging information (i.e., information that can be recorded on traditional paper). Thus, the document stores fixed-layout 2D static information that describes the visual appearance.

Software that implements a conforming implementation of the UOML specification is called a DoCbase Management System (DCMS). Applications process a UOML document by sending UOML instructions (operations) to the DCMS.

UOML defines an abstract document model and operations to that model. Examples of those operations include read/write, edit, display/print, query and security control. UOML covers operations that are required by many kinds of software applications in order to process documents.

UOML is based upon an XML schema, and is platform-independent, application-independent, programming language-independent, and vendor neutral. This standard will not restrict producers to implement a DCMS in a method of their choosing.

UOML allows different software applications to perform operations on the same document. A document can reside in the DCMS and applications can operate on that document. Those applications may have no relationship to each other besides the ability to send UOML instructions to the DCMS.

The UOML graphics object model is similar to the graphics model specified by ISO/IEC 32000-1:2008, the Portable Document Format (PDF) standard. For example, both standards describe a page layout using logical coordinate systems, and the positions of the graphics objects are specified using coordinates in the logical coordinate systems. The similarity of the two models allows UOML to be used as an interface standard for PDF.

This OASIS standard forms the foundation of UOML. Other standards building upon this standard may be created in the future.

End of informative text

(back to table)

NO-02:
Lack of Scope statement: Due to lack of scope statement the purpose of this standard is unclear.

Added a Scope clause to the specification.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

(back to table)

NO-05:
The explanation of "Docset" does not convey any meaning. In our interpretation a "directory in filesystem" is describing a hierarchical structure, containing documents. If a docset is is a set of documents distributed in a hierarchical structure this it should be described so.

Clarified the definition of both docbase and docset within the Terminology clause.

docbase: The root level of the UOML abstract document model. Abbreviated for "document base", it is the container of one or many documents. A docbase contains one and only one root docset. [*Note*: The docbase is analogous to a file system on a modern operating system. The term docbase is derived from the term "database". The docset is analogous to a directory within a file system on a modern operating system. The root docset is analogous to the root directory of a file system. *end note*].

docset: A set of documents. A docset may contain one to many docsets. [*Note*: The docset is analogous to a directory within a file system on a modern operating system. *end note*].

(back to table)

NO-06:
This definition does not convey any meaning. As written now a Path could be anything. It is also unclear what a “region collection” is.

Clarified the definition of Path within the Terminology clause.

Path: A Path is a graphics object composed of straight and/or curved line segments, which may or may not be connected. [*Note*: that in this document, 'path' (all lowercase) refers to a filename, location of docbase or image file. This is different from this current definition of “Path” (with the uppercase ‘P’). *end note*]

(back to table)

NO-07:
Several terms used within the definitions of the terminology in this document is not well defined, and should have a definition to be used.

Added a definition for “application software” within the Terminology clause. All other terms are self-explanatory within the context of their usage. “upper level object” has been stricken from the specification.

application software: Software that handles document data by issuing UOML instructions to the DCMS.

(back to table)

NO-08:
It is unclear if a Command object is used for modifying or “holding” graphics control parameters.

The definition of command object was rewritten for better clarity within the Terminology clause.

command object: An object used to modify the graphics state that holds the current graphics control parameters, such as text size, typeface and color.

(back to table)

NO-09:
It is unclear what the meaning of the 1st paragraph in the Overview is. And by reading the rest of the document, we have concerns on the validity of the statements put forward in the paragraph.

Rewrote the wording of this paragraph.

Overview

This clause is informative

This OASIS standard specifies an instruction set of XML elements and attributes describing operations on unstructured, fixed-layout documents. These instructions are for the processing of these documents to accomplish various functionality, such as display and edit.

UOML is to unstructured documents as SQL (Structured Query Language) is to structured data. UOML is expressed using standard XML via an instance of an XML schema. UOML handles fixed-layout documents and its associated information (e.g., metadata, security rights, etc.) Fixed-layout- documents are two-dimensional and contain static paging information (i.e., information that can be recorded on traditional paper). Thus, the document stores fixed-layout 2D static information that describes the visual appearance.

Software that implements a conforming implementation of the UOML specification is called a DoCbase Management System (DCMS). Applications process a UOML document by sending UOML instructions (operations) to the DCMS.

UOML defines an abstract document model and operations to that model. Examples of those operations include read/write, edit, display/print, query and security control. UOML covers operations that are required by many kinds of software applications in order to process documents.

UOML is based upon an XML schema, and is platform-independent, application-independent, programming language-independent, and vendor neutral. This standard will not restrict producers to implement a DCMS in a method of their choosing.

UOML allows different software applications to perform operations on the same document. A document can reside in the DCMS and applications can operate on that document. Those applications may have no relationship to each other besides the ability to send UOML instructions to the DCMS.

The UOML graphics object model is similar to the graphics model specified by ISO/IEC 32000-1:2008, the Portable Document Format (PDF) standard. For example, both standards describe a page layout using logical coordinate systems, and the positions of the graphics objects are specified using coordinates in the logical coordinate systems. The similarity of the two models allows UOML to be used as an interface standard for PDF.

This OASIS standard forms the foundation of UOML. Other standards building upon this standard may be created in the future.

End of informative text

(back to table)

NO-10:

Due to lack of Scope of this standard, we have problems understanding what is meant by this paragraph. Is this a standard for representing pixels in a "two dimensions" on a page. And if this standard is about "Static paging information" we have problems understanding what type of "UOML defined functions" an application can use to process this static document?

Added a scope clause to this standard and also modified the paragraph in question for better clarity.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

Also the Overview clause where this paragraph is found is informative only.

This OASIS standard specifies an instruction set of XML elements and attributes describing operations on unstructured, fixed-layout documents. These instructions are for the processing of these documents to accomplish various functionality, such as display and edit.

UOML is to unstructured documents as SQL (Structured Query Language) is to structured data. UOML is expressed using standard XML via an instance of an XML schema. UOML handles fixed-layout documents and its associated information (e.g., metadata, security rights, etc.) Fixed-layout- documents are two-dimensional and contain static paging information (i.e., information that can be recorded on traditional paper). Thus, the document stores fixed-layout 2D static information that describes the visual appearance.

Software that implements a conforming implementation of the UOML specification is called a DoCbase Management System (DCMS). Applications process a UOML document by sending UOML instructions (operations) to the DCMS.

(back to table)

NO-11:
It is unclear what is meant by this paragraph. The paragraph should be restructured to better explain what is meant.

The paragraph as it was has been removed and replaced with multiple paragraphs for better clarity.

Software that implements a conforming implementation of the UOML specification is called a DoCbase Management System (DCMS). Applications process a UOML document by sending UOML instructions (operations) to the DCMS.

UOML defines an abstract document model and operations to that model. Examples of those operations include read/write, edit, display/print, query and security control. UOML covers operations that are required by many kinds of software applications in order to process documents.

UOML is based upon an XML schema, and is platform-independent, application-independent, programming language-independent, and vendor neutral. This standard will not restrict producers to implement a DCMS in a method of their choosing.

UOML allows different software applications to perform operations on the same document. A document can reside in the DCMS and applications can operate on that document. Those applications may have no relationship to each other besides the ability to send UOML instructions to the DCMS.

(back to table)

NO-17:
Since the standard is lacking several important definitions, and since the Scope of the standard is not specified it is hard to make detailed comments on these parts. Based on the reading of the document, we find that there is a mix prosaic descriptions and illustrations which makes this document hard to interpret in a consistent way.

Added a scope, more illustrative examples, ensured the UOML XML schema (XSD) is part of the normative standard, etc. in order to make this a more clear standard.

Scope

This OASIS standard describes the abstract document model of UOML and the operations available on it. Specifically, operations providing functionality for read/write/edit and display/print on layout-based documents are described.

Here is an example of an example:

[*Example*:

Create a docbase, named 1.sep. If the DCMS successfully processed the OPEN instruction, it will return a RET instruction.

```
<OPEN path="/home/admin/storage/1.sep" create="true" del_exist="false"/>

Return element if OPEN succeeds:

<RET>
 <boolVal name="SUCCCESS" val="true"/>
 <stringVal name="HANDLE" val="db_handle_xxxxx"/>
</RET>

Return element if OPEN fails:

<RET>
 <boolVal name="SUCCCESS" val="false"/>
 <stringVal name="ERR_INFO" val="required resource not available"/>
</RET>
```

end example]

Normatively specified the UOML XSD

[UOMLSchema] *UOML Part 1 v1.0 Schema*, http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd

The formal definitions for these instructions are specified in the UOML XML Schema Definition located at: http://docs.oasis-open.org/uoml-x/v1.0/os/uoml-schema-os.xsd.

Also here is a business case for UOML:

The project "Electronic Document Workflow Platform Based on UOML" is used in the State Electricity Regulatory Commission of China, and has achieved good results. The State Electricity Regulatory Commission (SERC) is empowered by the State Council to perform administrative and regulatory duties with regard to the national electric power sector (in accordance with laws and regulations). It is responsible for the overall regulation of the national power sector, establishing a coherent system for regulatory organization, and exercising direct leadership over its regional branches.

The project "Electronic document workflow platform Based on UOML" in SERC integrated various UOML- conforming software products, including SURSEN electronic document transmission system, HANVON OCR software, the REDFLAG office suite software, the JETRICH handwriting software, FARSHIEN barcode identification system, TRS full-text indexing and UNIS archive management system. All these software products can open UOML-accessible documents, appending various elements, such as text, handwriting signature and two-dimensional barcodes. These products can also perform full-text search on UOML-accessible documents. As the products are all based on unified UOML interface, the document can interoperate among different software, so that the goal of a truly "paperless offices" is achieved finally with the office working environment quite efficient and convenient.

It is shown via this successful user case that several UOML-conformed products already exist from different vendors, and these products have been formally used in key applications.

The demonstration of this UOML user case can be found: http://www.uoml.org/flv04.html