
FAV[®] File Format Specification

作成者: 高橋智也¹⁾, 升森敦士²⁾, 藤井雅彦¹⁾, 田中浩也²⁾

¹⁾ 富士ゼロックス株式会社

²⁾ 慶應義塾大学 SFC ソーシャルファブリケーションラボ

発行日: 2018/MM/DD

バージョン: 1.1a

INDEX

序文.....	4
本仕様書の目的.....	4
本仕様書の記述.....	4
ライセンス.....	5
1. FAV ファイルフォーマット.....	6
1.1. コンセプト.....	6
1.2. 特徴.....	6
1.3. 環境条件.....	8
1.4. 構成.....	9
1.5. 記述方法.....	10
1.6. セキュリティ.....	10
2. <FAV>	11
3. <METADATA>.....	12
3.1. <ID>.....	12
3.2. <TITLE>.....	13
3.3. <AUTHOR>.....	13
3.4. <LICENSE>.....	13
3.5. <NOTE>.....	14
4. <PALETTE>.....	15
4.1. <GEOMETRY>.....	16
4.1.1. <shape>.....	17
4.1.2. <scale>.....	18
4.2. <MATERIAL>.....	19
4.2.1. <material_name>.....	20
4.2.2. <product_info>.....	20
4.2.3. <standard_name>.....	21
5. <VOXEL>.....	22
5.1. <GEOMETRY_INFO>.....	24
5.2. <MATERIAL_INFO>.....	25
5.3. <DISPLAY>.....	26
5.4. <APPLICATION_NOTE>.....	27
5.5. <REFERENCE>.....	28
6. <OBJECT>.....	30
6.1. <GRID>.....	32
6.1.1. <origin>.....	33
6.1.2. <unit>.....	33
6.1.3. <dimension>.....	34
6.2. <STRUCTURE>.....	34
6.2.1. <voxel_map>.....	35
6.2.2. <color_map>.....	37
6.2.3. <link_map>.....	39
6.2.4. <user_defined_map>.....	42
6.2.4.1. ユーザー定義属性ファイル - XML 形式 (.favmapx ファイル).....	43
6.2.4.2. ユーザー定義属性ファイル - Binary 形式 (.favmap ファイル).....	45
7. まとめ.....	47
8. 変更履歴.....	61

序文

本仕様書の目的

本仕様書は、FAV (FAbricatable Voxel) ファイルフォーマットの仕様について説明する。
本仕様書は、以下の目的を実現するために公開される。

- ・ 3D モデルデータの内部まで含めた形状や属性などの複雑な構造を FAV フォーマットで効率的に管理する
- ・ FAV フォーマットにより管理された 3D モデルデータの情報をハードウェアやソフトウェアで活用する
- ・ FAV フォーマットの作成、編集とシステムによる利活用を相互運用する

本仕様書に則ってフォーマットされた 3D モデルデータは、FAV ファイルを扱うことができるシステムによって利用することができる。FAV ファイルを処理するシステムは、本仕様書の記載に則り適切に処理することで、FAV フォーマットで管理される 3D モデルデータの様々な情報を利用することができる。

我々は、FAV フォーマットを利用することで、3D モデルデータのデザイン性や表現力が向上し、3D モデルデータを利用したものづくり (fabrication) やコミュニケーションが発展していくことを望んでいる。

本仕様書の記述

● RFC2119

RFC2119 に準拠した表現で記載する。以下に、RFC2119 を転載する。

1. 「しなければならない (MUST)」
この語句、もしくは「要求されている (REQUIRED)」および「することになる (SHALL)」は、その規定が当該仕様の絶対的な 要請事項であることを意味します。
2. 「してはならない (MUST NOT)」
この語句、もしくは「することはない (SHALL NOT)」は、その規定が当該仕様の絶対的な禁止事項であることを意味します。
3. 「する必要がある (SHOULD)」
この語句もしくは「推奨される (RECOMMENDED)」という形容表現は、特定の状況下では、特定の項目を無視する正当な理由が存在するかもしれませんが、異なる選択をする前に、当該項目の示唆するところを十分に理解し、慎重に重要性を判断しなければならない、ということの意味します。
4. 「しないほうがよい (SHOULD NOT)」
この語句もしくは「推奨されない (NOT RECOMMENDED)」という形容表現は、特定の動作が容認できる、ないし、非常に有用である、というような特定の状況下では、正当な理由が存在するかもしれませんが、このレベルの動作を実装する前に、当該項目の示唆するところを十分に理解し、慎重に重要性を判断しなければならない、ということの意味します。
5. 「してもよい (MAY)」
この語句、もしくは「選択できる (OPTIONAL)」という形容表現は、ある要素が、まさに選択的であることを意味します。その要素を求めている特定の市場があるから、あるいは、他のベンダーはその要素を提供しないだろうが、その製品機能を拡張する と察知して、その要素を含む選択をするベンダーがあるかもしれません。特定の選択事項 (オプション) を含まない実装は、おそらく機能的には劣る ことになるでしょうが、そのオプションを含む他の実装との相互運用に備えなければなりません (MUST)。同様に、特定のオプションを含む実装は、そのオプションを含まない実装 との相互運用に備えなければなりません (MUST)。
(当然ながら、そのオプションが提供する機能は除かれます。)

本仕様書中で上記 1~5 の表現を使用して記載された仕様は、FAV フォーマットの目的や意図通りにシステムを相互運用するための必要条件である。これに反して FAV フォーマットを使用する場合、FAV フォーマットを利用するシステムに重大な問題を引き起こす可能性があることを十分に理解しなければならない。

仕様に反する FAV フォーマットを利用することは、システムに適切なエラー処理が実装されていることを要求することであり、それが確実に保証されるものではない。FAV フォーマットを利用するシステムに意図しない動作を引き起こさないように、十分注意する必要がある。

● XML

FAV フォーマットは、XML により記述される。

本仕様書の本文中では、XML 内の Element を Angle brackets で囲んだ形式で記載する。 (例) <element>
Element の Attribute を Parentheses で囲んだ形式で記載する。 (例) (attribute)

本仕様書では、XML の記載例は四角で囲み、(例)というラベルを添えて記載される。

(例) <element_label attribute_label=0.1>ELEMENT</element_label>

また、本仕様書の element, 並びに attribute に関する説明は以下のように表形式で記載する。

表 1: element, 並びに attribute 仕様の記載例

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
element_label	attribute_label	Double	Double 値 [0.0]	Element についている Attribute のデータに何を指定するか、どんな意味があるのかを説明する。	必須, 1 つ以上, 等の条件を指定する
	-	String	String 値 [-]	Element のデータに何を指定するのか、どんな意味があるのかを説明する。デフォルト値がない場合は“-”と記載する。	

本仕様書の本文中では、Angle brackets で囲んだ Element をドット(.)で区切って記載した場合、ドットの左側を親階層、右側を子階層という意味として使用する。

(例) <element_parent.element_child>は、以下の Element を示すための記載方法として使用する。

```
<element_parent>
  <element_child>この element のことを示すための記載であるとして使用する。 </element_child>
</element_parent>
```

ライセンス

本仕様書の著作権者およびライセンスは以下の通りとする。

ライセンスに関する表示事項:

© 2018, Fuji Xerox Co., Ltd. , and Dr. Hiroya Tanaka,
クリエイティブ・コモンズ 表示-改変禁止 4.0 国際
<<https://creativecommons.org/licenses/by-nd/4.0/>>

これは、本仕様書自体の複製や改変に関わるライセンスであり、FAV フォーマットに則って定義された個別のデータや、FAV フォーマットを利用するシステムについて、利用を制限したり、許諾したり、権利を主張するものではない。

本仕様書の全部、または一部を複製その他の形で利用する場合は、以下のような記載をすることで、ライセンスに規定されている記載義務を満たすことができる。

- ・ 上記「ライセンスに関する表示事項」
- ・ 仕様書の URL
- ・ 「FAV File Format Specification Ver 1.1 の一部の複製」などの明記

1. FAV ファイルフォーマット

FAV ファイルフォーマットは、3D モデルデータを保存するためのファイルフォーマットである。

1.1. コンセプト

FAV フォーマットは、FAbricatable Voxel フォーマットの略である。

FAV フォーマットでは、3D モデルデータの外部形状だけでなく、内部構造、使用する材料、色、接合強度などの様々な属性を保持することができる。3D モデルの外側/内側を問わず、デザイナーが思った通りに・隅々まで・徹底的に・精緻にデザインできること、それをデータとして保存できることを担保している。

3D モデルデータにより高い表現力を発揮し、ものづくりに活かすことで、これまで作ることができなかったものを考え、作ることができ、3D モデルデータを介したコミュニケーションの促進にも寄与する。特に、これまで分かれていた、画像処理、医療用の MRI/CT スキャンデータ、複合材料の混合による工業製品といった3つの分野を横断し、相互の境界を超えた新しいアプリケーションを生み出すことに役立つ。

FAV フォーマットで記載された 3D モデルデータは、fabricatable である、つまりものづくりに利用するために最適化された情報として格納されていることを示している。ものづくりに利用することができる最適化された情報とは、以下のようなことを示す。

- ・ 3D モデルデータの外部、内部を問わず、形状、材料、色、接合強度、等といったものづくりに必要な情報が立体的な位置ごとに明確に定義されていること
- ・ 3D モデルデータのデザイン(CAD)、解析(CAE)、検査(CAT)を、データ変換することなく統一的、かつ双方向的に行うことができること

FAV フォーマットは、これらを兼ね備えた 3D モデルデータである。

1.2. 特徴

FAV フォーマットは、FAbricatable Voxel の略であり、ボクセルをベースに構成されている。

ボクセルとは、3 次元的な画素値である。2 次元的な画素値である pixel を平面的に配置することで、画像を構成するように、3 次元的な画素値である voxel を立体的に配置することで、物体を構成する。

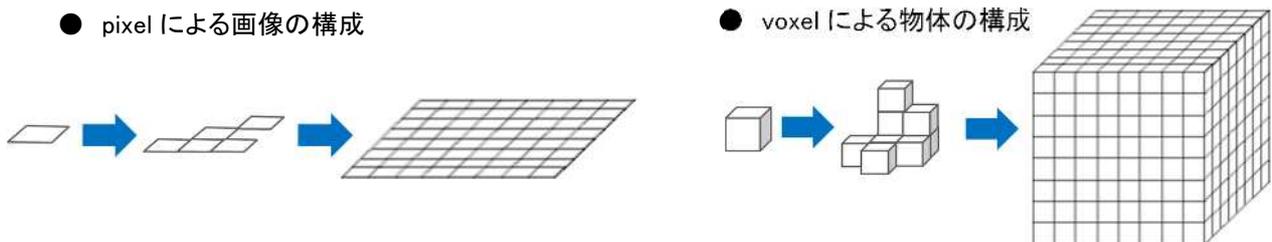


図 1 : pixel と voxel

Voxel を用いて 3 次元的な形状を定義することができるため、種類の違う Voxel を組み合わせたり、Voxel の有り無しを組み合わせたりすることで、様々な構造を表現することができる。

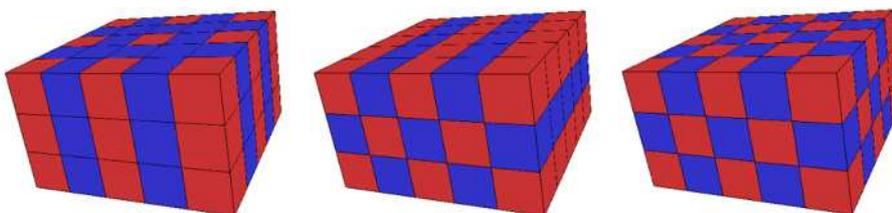
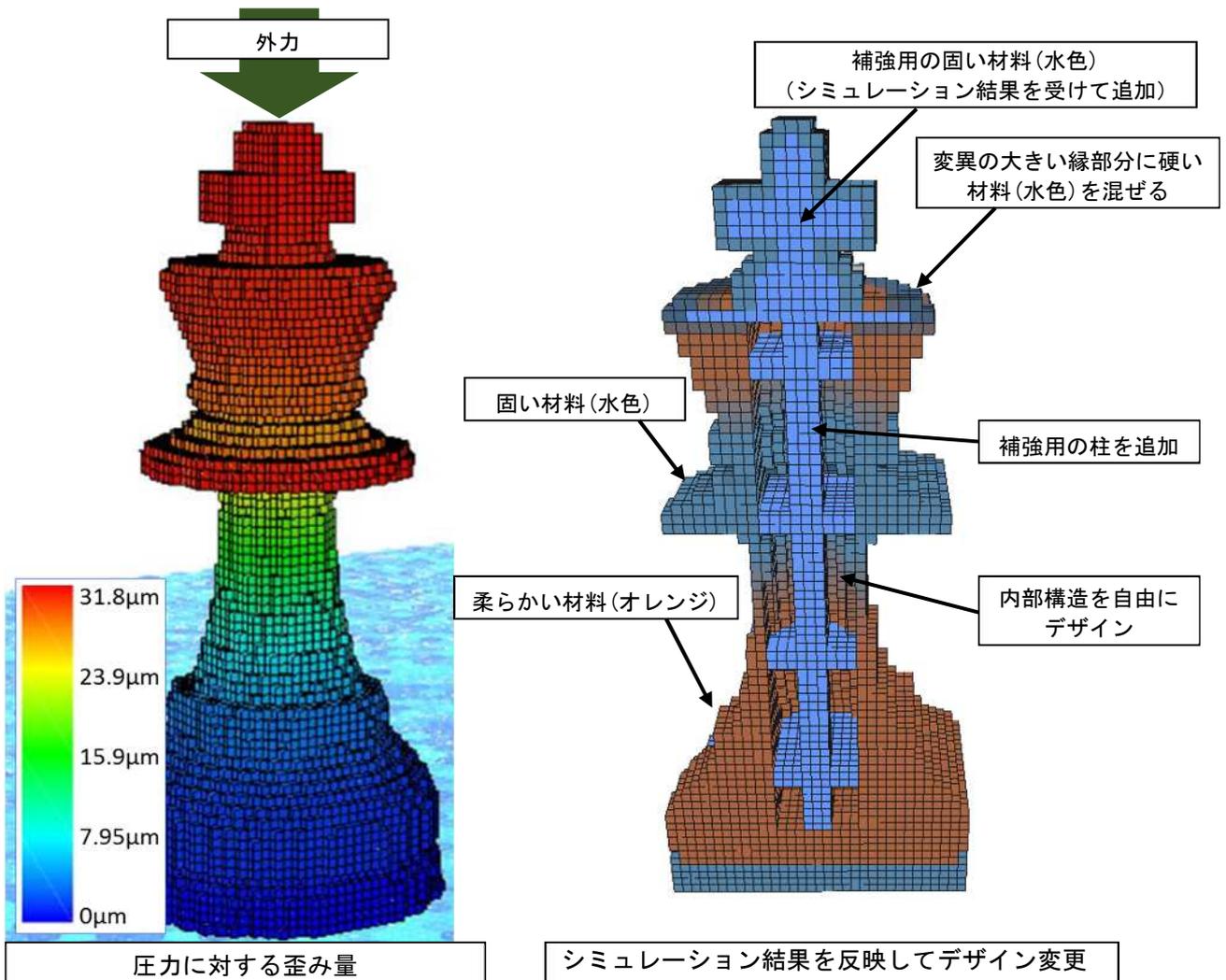


図 2 : Voxel の組み合わせによる微細構造の表現

Voxel という基本要素の積み上げで 3 次元的な物体を定義することで、以下のようなメリットを得ることができる。

- ・ 外側から見た形状だけでなく、内側の形状まで Voxel の配置で表現できる
- ・ 形状だけでなく、材料や色や接合強度などといった様々な属性の構造を Voxel の配置で表現できる
- ・ Voxel は有限要素であり、デザイン(CAD)したフォーマットのままシミュレーション(CAE)を行うことができる。また、シミュレーション結果から voxel 単位での設計変更を行うことができる。



シミュレーション結果を反映でき、設計変更が容易

- ・ 材料変更 (材料を固くする。など)
- ・ 構造変更 (形を補強、内部を省略、など)

図 3 : voxel で 3D モデルデータを構成するメリット

1.3. 環境条件

FAV フォーマットは、以下のような環境であることを前提としてデータが保持される。

● 座標系

FAV フォーマットで扱われる座標系は、積層面を XY 平面とし、積層高さ方向を Z 軸プラス方向であるとする、右手座標系であることが要求されている。

3D プリンターなどの製造装置のワークエリアを考えた時、装置の原点から見て製造する高さが Z 軸、幅が X 軸、奥行きが Y 軸となるように向きを合わせ、3D モデルデータが収まるように配置することが推奨される。

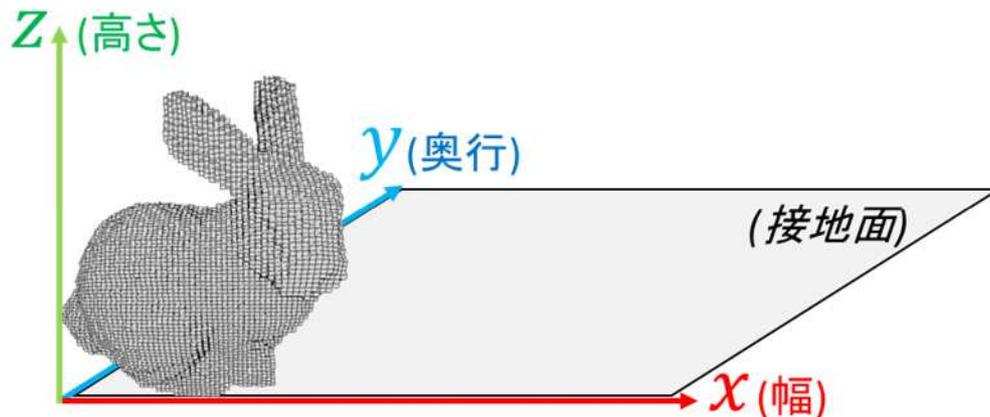


図 4 : FAV フォーマットで使用される座標系

● 座標の単位

FAV フォーマット内で使用される座標値 1 は、実世界上の 1mm であることが要求されている。

アプリケーションによって、インチや他の単位に変換して使用してもよい。ただしその場合でも、FAV フォーマットに保存する際は、座標値 1 が実世界上の 1mm になるようにしなければならない。環境やアプリケーションによって、同じ FAV ファイルが違う寸法で解析されることを避けなければならない。

1.4. 構成

FAV フォーマットの各種要素は、以下のようなツリー構造を持つ。

FAV フォーマットには、大きく分けて 4 つのパート<metadata>, <palette>, <voxel>, <object>がある。<palette>は<voxel>から参照され、<voxel>は<object>から参照される。そのため、FAV フォーマットの先頭から<palette>→<voxel>→<object>の順に定義することが推奨される。その他、特に記載がない限り、要素の順序には規定がないものとする。

以下、それぞれの要素、属性の詳細な説明、及び条件は各節を参照。

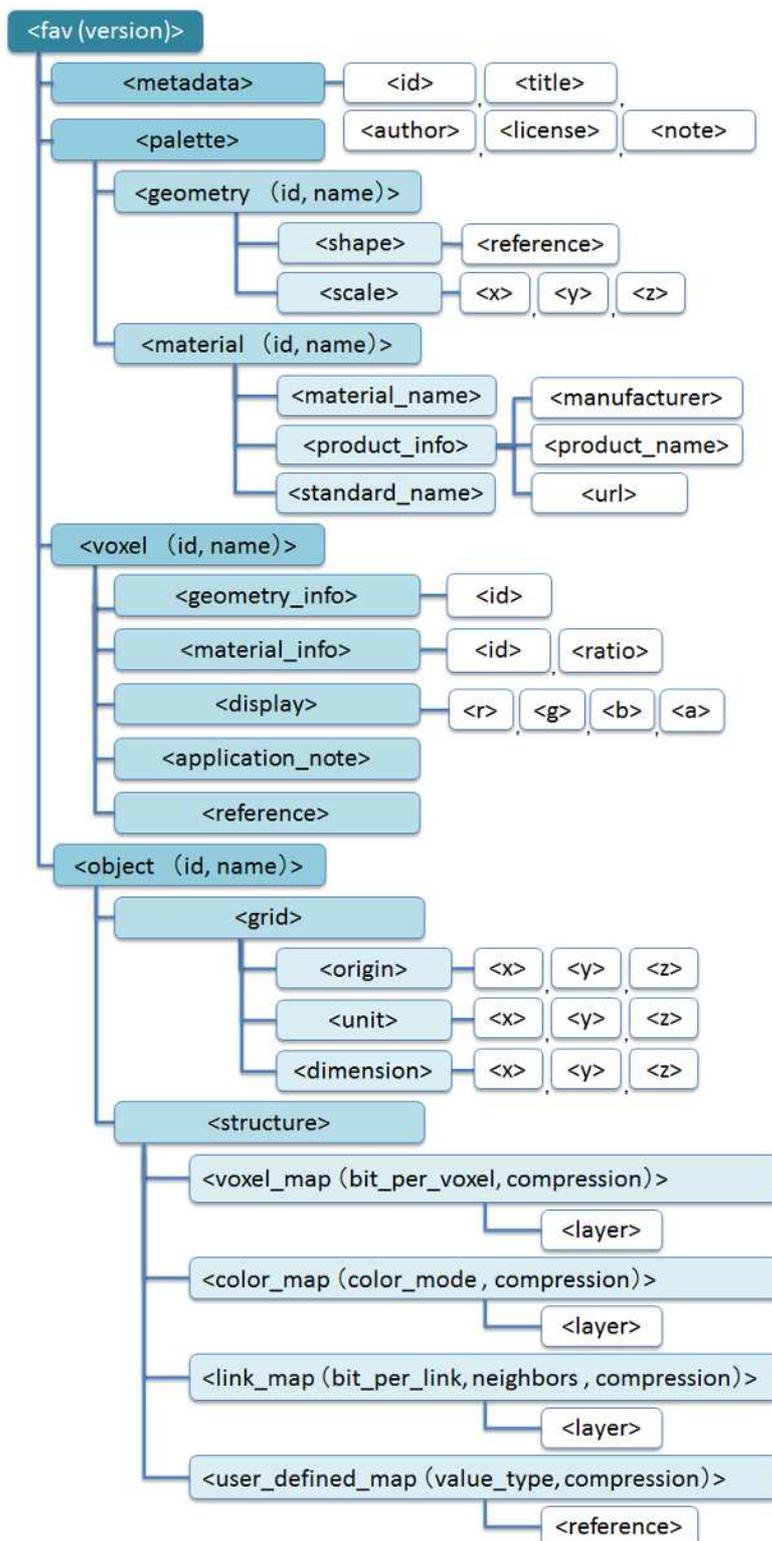


図 5 : fav を構成する各要素のツリー構造

1.5. 記述方法

FAV フォーマット内の要素, 並びに属性のラベルは, すべて小文字で記載することになる.
複数の単語からなるラベルは, 単語同士が underline で接続されて記載することになる.

(例) <element_item attribute_item="ATTRIBUTE"/>

1.6. セキュリティ

FAV フォーマットに対するセキュリティは, 他の一般的な電子データと同様であるとして扱う. つまり, パスワードを設定する, 改ざんを検知するための電子署名を付与する, 特定の鍵を保持している場合のみ FAV ファイルを閲覧できるようにする, 等といったセキュリティ技術を適用することができる.

作者が許諾していない改編や, 作者情報そのものの改変などを行った FAV フォーマットを再流通してはならない.
ライセンスに反する利用を行った場合, 法律に違反する可能性があることを十分に理解しなければならない.

2. <fav>

親要素: — , リンク: <metadata>, <palette>, <voxel>, <object>

<fav>は, FAV フォーマットの Root 要素である. FAV フォーマットに則って定義された 3D モデルデータは, <fav>という要素から開始される.

FAV フォーマットで定義され, 管理されるデータはすべて<fav>以下に格納される.

<fav>は以下の属性を持つ.

表 2: <fav>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
fav	version	String	FAV フォーマットのバージョン番号 [—]	この FAV ファイルが則しているフォーマットのバージョン番号を文字列で指定する.	必須

<fav>以下の階層には, 以下の要素を記載してもよい. 詳細は<metadata>節で説明する.

- <metadata>

以下に, <fav>以下の階層の XML のサンプルを示す.

```
(例)
<fav version="1.1a">
...
</fav>
```

図 6 : <fav>以下の階層の XML サンプル

3. <metadata>

親要素: <fav>, リンク: <material>, <object>, <user_defined_map>

<metadata>は、FAV フォーマットで定義される各種データに関するメタデータが記載される。

<metadata>は、<fav>, <material>, <object>, <user_defined_map>を親要素として定義することができる。どの階層に定義した場合でも、<metadata>以下に定義できる要素は同じである。ただし、必要な要素のみ定義すればよい。

<metadata>には、以下の要素を定義する。

- <id>
- <title>
- <author>
- <license>
- <note>

以下に、<metadata>以下の階層の XML のサンプルを示す。

```
(例)
<metadata>
  <id>bc4affb5-9a53-4de7-9f27-721ef27e8f34</id>
  <title><![CDATA[FAV Ver1.1 Sample File]]></title>
  <author><![CDATA[Fuji Xerox & Keio SFC]]></author>
  <license><![CDATA[CC BY]]></license>
  <note><![CDATA[This is a sample file in FAV format ver1.1.]]></note>
</metadata>
```

図 7 : <metadata>以下の階層の XML サンプル

3.1. <id>

親要素: <metadata>, リンク: <fav>, <material>, <object>, <user_defined_map>

<id>は、<metadata>を定義する親要素のデータを一意に示す ID 情報である。FAV フォーマットで管理されるデータの一意性を担保しようとする場合、GUID などの一意性が担保されている ID 情報を使用することが推奨される。

<id>は以下のデータを持つ。

表 3:<id>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
id	—	String	親要素の ID [—]	<metadata>を定義する親要素を一意に示す ID を文字列で指定する。	必須

<id>の XML のサンプルは図 7 を参照。

3.2. <title>

親要素: <metadata>, リンク: <fav>, <material>, <object>, <user_defined_map>

<title>は, <metadata>を定義する親要素を表す名前である. <fav.metadata.title>であれば<fav>全体を表す名前であり, <object.metadata.title>であれば1つの<object>を表す名前である.

<title>は以下のデータを持つ.

表 4: <title>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
title	—	CDATA	親要素の名前 [—]	<metadata>を定義する親要素を表す名前を文字列で指定する.	必須

<title>のXMLのサンプルは図 7を参照.

3.3. <author>

親要素: <metadata>, リンク: <fav>, <material>, <object>, <user_defined_map>

<author>は, <metadata>を定義する親要素の作者情報である. <fav.metadata.author>であれば<fav>全体の作者であり, <object.metadata.author>であれば1つの<object>の作者である.

<author>は以下のデータを持つ.

表 5: <author>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
author	—	CDATA	親要素の作者 [—]	<metadata>を定義する親要素の作者情報を文字列で指定する.	必須

<author>のXMLのサンプルは図 7を参照.

3.4. <license>

親要素: <metadata>, リンク: <fav>, <material>, <object>, <user_defined_map>

<license>は, <metadata>を定義する親要素のライセンス情報である. <fav.metadata.license>であれば<fav>全体のライセンスを表し, <object.metadata.license>であれば1つの<object>のライセンスを表す.

ライセンスには, Creative Commons(CC), GNU General Public License(GPL), BSD, X11(MIT), 等が存在する. また, 独自ライセンスを定義する場合, <license>以下に全文を記載するほか, ライセンス情報へのリンクを記載する, 等を行う.

<license>は以下のデータを持つ.

表 6: <license>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
license	—	CDATA	親要素のライセンス情報 [—]	<metadata>を定義する親要素のライセンス情報を文字列で指定する.	必須

<license>のXMLのサンプルは図 7を参照.

3.5. <note>

親要素: <metadata>, リンク: <fav>, <material>, <object>, <user_defined_map>

<note>は, <metadata>を定義する親要素に関するメモである. <fav.metadata.note>であれば<fav>全体に関するメモであり, <object.metadata.note>であれば1つの<object>に関するメモである.

<note>は以下のデータを持つ.

表 7: <note>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
note	—	CDATA	親要素に関するメモ [—]	<metadata>を定義する親要素に関するメモを文字列で指定する.	

<note>のXMLのサンプルは図 7を参照.

4. <palette>

親要素: <fav>, リンク: <voxel>, <geometry>, <material>

<palette>では、3D モデルデータを FAV フォーマットに基づいて構成するための前準備として、ボクセルの形状や材料などの基本情報の登録を行う。<palette>に登録された基本情報で構成される<voxel>を用いて<object>を定義することで、3D モデルデータを FAV フォーマットとして定義する。

<palette>以下の階層には、以下の要素を記載する。

- <geometry>
- <material>

以下に、<palette>以下の階層の XML のサンプルを示す。

```
(例)
<palette>
  <geometry id="1" name="NormalCube">
    <shape>cube</shape>
    <scale>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </scale>
  </geometry>
  <geometry id="2" name="Plate">
    <shape>cube</shape>
    <scale>
      <x>1</x>
      <y>1</y>
      <z>0.25</z>
    </scale>
  </geometry>
  <geometry id="3" name="Diamond">
    <shape>user_defined</shape>
    <reference><![CDATA[Diamond.stl]]</reference>
    <scale>
      <x>0.98</x>
      <y>0.98</y>
      <z>-1.05</z>
    </scale>
  </geometry>
  <material id="1" name="SoftMat1">
    <material_name><![CDATA[Some-soft-materials]]</material_name>
  </material>
  <material id="2" name="HardMat1">
    <product_info>
      <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
      <product_name><![CDATA[ULTRA-HARD/007]]</product_name>
      <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]</url>
    </product_info>
    <product_info>
      <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
      <product_name><![CDATA[ULTRA-HARD/006a]]</product_name>
      <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]</url>
    </product_info>
  </material>
</palette>
```

```

    <standard_name><![CDATA[JIS K6899-1 ABS]]>
  </standard_name>
</material>
</palette>

```

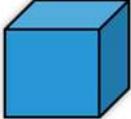
図 8 : <palette>以下の階層の XML サンプル

4.1. <geometry>

親要素: <palette>, リンク: <voxel>, <shape>, <scale>

<geometry>は、3D モデルデータを構成する基本要素となるボクセルの形状や倍率の定義である。ここで定義した geometry のボクセルを立体的に積み重ねることで、3D モデルデータを構成する。

表 8:<geometry>のパターン例

イメージ						
(id)	01	02	03	04	05	06
(name)	Cube01	Cube02	Plate	BigSphere	SmallSphere	Cylinder
<shape>	cube	cube	cube	sphere	sphere	user_defined
<scale>	2 × 2 × 2	1 × 1 × 1	1 × 1 × 0.3	1.5 × 1.5 × 1.5	0.25 × 0.25 × 0.25	3 × 1 × 1

<geometry>は以下の属性を持つ。

表 9:<geometry>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
geometry	id	positive Integer	幾何情報の ID [1 [*]] [*] 以降、1 ずつ加算	<geometry>を一意に示すための ID を設定する。他の<geometry>の(id)と重複してはならない。	必須
	name	String	定義する幾何情報の定義名 [-]	<geometry>の名前を設定する。他の<geometry>の(name)と重複しない方が良い。	

<geometry>以下の階層には、以下の要素が記載される。

- <shape>
- <scale>

以下に、`<geometry>`以下の階層の XML のサンプルを示す。

```
(例)
<geometry id="2" name="Plate">
  <shape>cube</shape>
  <scale>
    <x>1</x>
    <y>1</y>
    <z>0.25</z>
  </scale>
</geometry>
<geometry id="3" name="Diamond">
  <shape>user_defined</shape>
  <reference><![CDATA[Diamond.stl]]</reference>
  <scale>
    <x>0.98</x>
    <y>0.98</y>
    <z>-1.05</z>
  </scale>
</geometry>
```

図 9 : `<geometry>`以下の階層の XML サンプル

4.1.1. `<shape>`

親要素: `<geometry>`, リンク: `<voxel>`, `<grid>`

`<shape>`は、3D モデルデータを構成する基本要素となるボクセルの形状である。既定の形状の他、外部の stl ファイルを指定して任意の形状を使用することができる。

指定した`<shape>`で構成される`<voxel>`を`<grid>`に配置した場合、`<shape>`の中心が`<grid>`セルの中心と一致するようにセンタリングされて配置される。

`<shape>`は、以下のデータを持つ。

表 10: `<shape>`の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
shape	—	String	cube / sphere / user_defined [cube]	Voxelの形状を指定する。user_definedを指定した場合、 <code><reference></code> を追加で指定しなければならない。	

`<shape>`に user_defined を指定した場合、ボクセルの形状を定義する外部の stl ファイルへの参照を示す`<reference>`を追加で定義しなければならない。

表 11: `<reference>`の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
reference	—	CDATA	ボクセルの形状を定義する stl ファイルパス [—]	Voxelの形状を定義する。外部の stl ファイルまでの相対パスを文字列で指定する。	

`<shape>`の XML のサンプルは図 9 を参照。

4.1.2. <scale>

親要素: <geometry>, リンク: <voxel>, <shape>, <grid>

<scale>は, 3D モデルデータを構成する基本要素となるボクセルの倍率である. 倍率は, ボクセルを配置する空間である<grid>のセルに対する倍率を指定する. <grid>のセルと等倍の $1 \times 1 \times 1$ の倍率が基本となる.

指定した<scale>で構成される<voxel>を<grid>に配置した場合, <scale>で指定した倍率のかかった<shape>の中心が<grid>セルの中心と一致するようにセンタリングされて配置される.

<scale>は以下の要素を持つ.

表 12: <scale>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
x	—	Double	X 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の X 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	
y	—	Double	Y 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の Y 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	
z	—	Double	Z 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の Z 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	

<scale>の XML のサンプルは図 9 を参照.

4.2. <material>

親要素: <palette>, リンク: <voxel>

<material>は、3D モデルデータを構成する基本要素となるボクセルに設定する材料情報である。ここで定義した material のボクセルを立体的に積み重ねることで、3D モデルデータの材料構成を定義する。

<material>は、以下の属性を持つ。

表 13: <material>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
material	id	positive Integer	材料情報の ID [1*] ※以降、1 ずつ加算	<material>を一意に示すための ID を設定する。他の <material>の(id)と重複してはならない。	必須
	name	String	定義する材料情報の名称 [-]	<material>の名前を設定する。他の<material>の(name)と重複しない方が良い。	

<material>以下の階層には、材料情報を表す以下の要素をいずれか 1 つ以上定義する。

- <material_name>
- <product_info>
- <standard_name>

これらの要素は、指定された材料をユーザーが使用不可能であった場合に備え、<material>以下の階層に複数定義することができる。その場合、使用を推奨する順番で要素を定義する。

また、<material>以下の階層に、以下の要素を記載してもよい。詳細は<metadata>節で説明する。

- <metadata>

以下に、<material>以下の階層の XML のサンプルを示す。

```
(例)
<material id="2" name="HardMat1">
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
    <product_name><![CDATA[ULTRA-HARD/007]]</product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]</url>
  </product_info>
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
    <product_name><![CDATA[ULTRA-HARD/006a]]</product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]</url>
  </product_info>
  <standard_name><![CDATA[JIS K6899-1 ABS]]>
</standard_name>
</material>
```

図 10 : <material>以下の階層の XML サンプル

4.2.1. <material_name>

親要素:<material>, リンク: -

<material_name>では, 材料を名前で指定する. 一般的に使用されている ABS, PLA 等の材料名のほか, 商品名や通称など, なんの材料を使用するかを判別することができる文字列を指定する.

<material_name>は以下のデータを持つ.

表 14: <material_name>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
material_name	-	CDATA	材料の名前情報 [-]	なんの材料を使用するかを判別することができる文字列を指定する.	

<material_name>の XML のサンプルは図 10 を参照.

4.2.2. <product_info>

親要素:<material>, リンク: -

<product_info>では, 材料を製品情報で指定する. メーカー名, 商品名, または商品コード, 商品情報 URL など, なんの材料を表しているかを判別することができる製品情報を指定する.

<product_info>は以下の要素を持つ.

表 15: <product_info>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
manufacturer	-	CDATA	材料メーカー名 [-]	材料のメーカー名を文字列で指定する.	
product_name	-	CDATA	材料の製品名 [-]	材料を表すメーカー指定の商品名, 商品コード, 等を文字列で指定する.	
url	-	CDATA	材料に関する web 上の情報の url [-]	材料に関する web 上の情報へのリンクを url で指定する.	

<product_info>の XML のサンプルは図 10 を参照.

5. <voxel>

親要素: <fav>, リンク: <geometry>, <material>, <display>, <application_note>, <reference>

<voxel>は、3D モデルデータを FAV フォーマットに基づいて構成するための基本要素であるボクセルを定義する。ボクセルを立体的に積み上げていくことで、3D モデルデータの構造を定義する。<voxel>は<geometry>, <material>などの情報を保持する。そのため、3D モデルデータの各位置において、形状以外の情報も定義することができる。

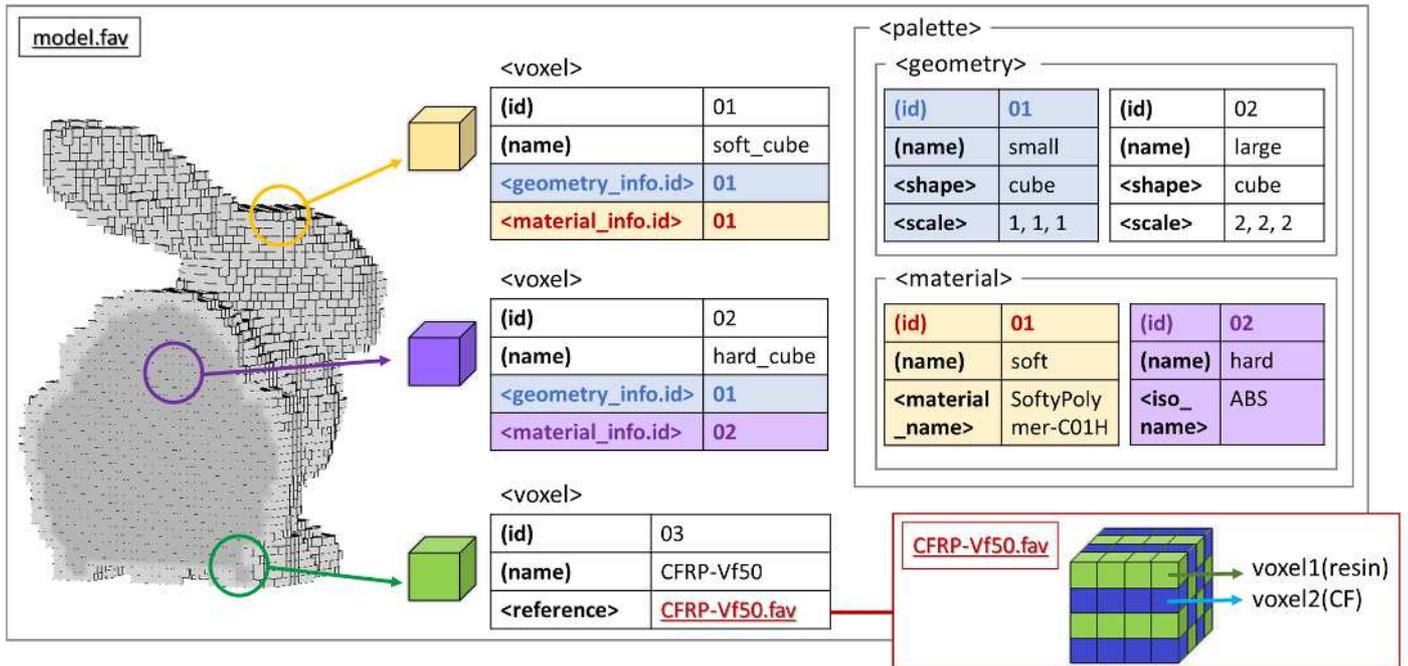


図 11 : voxel による形状と属性の指定の例

<voxel>は以下の属性を持つ。

表 17: <voxel>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
voxel	id	positiveInteger	ボクセルの ID [1*] *以降、1 ずつ 加算	<voxel>を一意に示すための ID を 1 以上の整数で設定しなければならない。他の<voxel>の(id)と重複してはならない。	必須
	name	String	定義するボクセルの名称 [-]	<voxel>の名前を設定する。他の<voxel>の(name)と重複しない方が良い。	

<voxel>の(id)は、3D モデルデータの形状を定義するための<voxel_map>にて、16 進数文字列として列挙されるため、整数であることが要求されている。また、0 は<voxel>が存在しないことを示すために reserve されており、(id)に使用してはならない。

<voxel>以下の階層には、以下の要素を記載する。または、<reference>のみを定義することで、外部の fav ファイルを指定して任意のボクセル集合を 1 ボクセルとして使用することができる。

- <geometry_info>
- <material_info>
- <display>
- <application_note>

以下に、<voxel>以下の階層のXMLのサンプルを示す。

```

(例)
<voxel id="1" name="soft_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>1</ratio>
  </material_info>
</voxel>
<voxel id="2" name="hard_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>0.15</ratio>
  </material_info>
  <material_info>
    <id>2</id>
    <ratio>0.85</ratio>
  </material_info>
  <application_note><![CDATA[HM-H01:Hybrid Hard Material Number 01]]></application_note>
  <application_note><![CDATA[FabAppAttr : application note]]></application_note>
</voxel>
<voxel id="3" name="reserved_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>0</id>
    <ratio>1</ratio>
  </material_info>
</voxel>
<voxel id="4" name="sparse_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>2</id>
    <ratio>0.6</ratio>
  </material_info>
  <material_info>
    <id>0</id>
    <ratio>0.4</ratio>
  </material_info>
</voxel>
<voxel id="5" name="Voxel01">
  <reference><![CDATA[Voxel01.fav]]></reference>
</voxel>

```

図 12 : <voxel>以下の階層のXMLサンプル

5.1. <geometry_info>

親要素: <voxel>, リンク: <palette>, <geometry>

<geometry_info>では, <voxel>に対して形状や倍率を示す<geometry>を指定する.

<geometry_info>の<id>には, <palette>に登録されている<geometry>の(id)を指定する. この<geometry_info>が設定されている<voxel>によって 3D モデルデータの形状を定義した場合, 該当する箇所の単位形状が指定された<geometry>であることを示す (図 11) .

<geometry_info>は以下の要素を持つ.

表 18: <geometry_info>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
id	—	positiveInteger	定義済みの形状情報の ID [—]	<geometry>を参照するための ID を指定する. <geometry>に設定済みの(id)を指定しなければならぬ.	必須

<geometry_info>の XML のサンプルは図 12 を参照.

5.2. <material_info>

親要素: <voxel>, リンク: <palette>, <material>

<material_info>では、<voxel>に対して材料情報を示す<material>を設定する。

<material_info>の<id>には、<palette>に登録されている<material>の<id>, もしくは 0 を指定する。

<palette>に登録されている<material>の<id>を指定した場合、そのボクセルが配置された箇所を構成する材料が指定された<material>であることを示す (図 11)。

0 を指定した場合、そのボクセルが配置された箇所には材料がない空間が確保されていることを示す。

<material_info>を<voxel>以下の階層に複数定義することで、複合材料を定義することができる (図 13)。その場合、<material_info>には、<material>の種類を指定する<id>と、複合する割合を指定する<ratio>のペアを指定する。一つの<voxel>に指定されたすべての<material_info>の<ratio>の合計が 1.0 になるよう指定する必要がある。

<id>が 0 の<material_info>を組み合わせた場合、1 つのボクセル内に、指定された<ratio>の割合分、材料がない空間が含まれていることを示す。

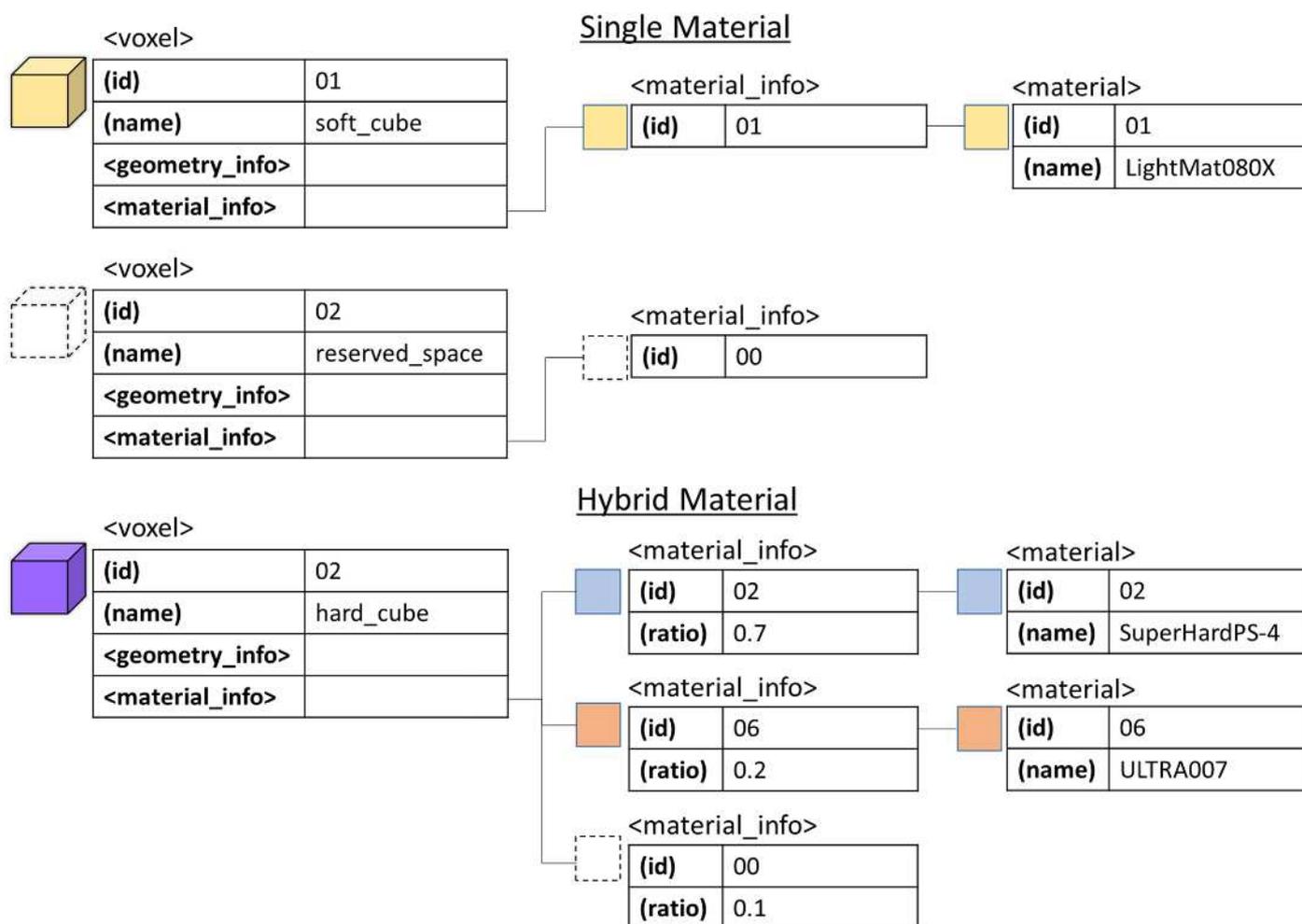


図 13 : 単一材料と複合材料の定義例

<material_info>は以下の要素を持つ。

表 19: <material_info>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
id	—	positiveInteger	定義済みの材料情報の ID [—]	<material>を参照するための ID を指定する。<material>に設定済みの(id), もしくは 0 を指定しなくてはならない。	必須
ratio	—	Double	複数の材料の複合割合 [—]	<material>を使用する割合を Double 値で指定する。他の<material>の割合と合計で 1.0 になるように指定する必要がある。0 以下を指定してはならない。	

<material_info>の XML のサンプルは図 12 を参照。

5.3. <display>

親要素: <voxel>, リンク: <color_map>

<display>は、<voxel>を判別するための色情報を表す。形状や材料などの属性の違いを分かりやすく可視化するために使用される。3D モデルデータの形状定義時、<voxel>に<display>を指定した場合、3D モデルデータの指定された位置の色は<display>の色で表示する必要がある。

3D モデルデータを実際にものとして作り出す際、ものの色として利用する情報は、<color_map>を使用することになる。

<display>は以下の要素を持つ。

表 20: <display>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
r	—	nonNegative Integer	ボクセル表示色の赤 (Red) 成分 [—]	<voxel>の色情報を表す Red 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	
g	—	nonNegative Integer	ボクセル表示色の緑 (Green) 成分 [—]	<voxel>の色情報を表す Green 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	
b	—	nonNegative Integer	ボクセル表示色の青 (Blue) 成分 [—]	<voxel>の色情報を表す Blue 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	
a	—	nonNegative Integer	ボクセル表示色の透過度 [—]	<voxel>の色情報を表す Alpha 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	

<display>の XML のサンプルは図 12 を参照。

5.4. <application_note>

親要素: <voxel>, リンク: —

<application_note>では, <voxel>に格納する任意のプロパティ情報を定義する.

格納したいプロパティ1 つにつき1 つの<application_note>要素を追加して指定する. <application_note>のデータには, プロパティを格納したユーザー, およびプロパティを使用するユーザーが複数の<application_note>の中から必要なデータを判別・抽出できるような, 手がかりとなる記載を添えておくことが推奨される.

サイズの大きなデータを格納することや, 不必要に多くの<application_note>を定義することは推奨されない.

<application_note>は以下の要素を持つ.

表 21: <application_note>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
application_note	—	CDATA	ボクセルに格納する任意のプロパティ情報 [—]	<voxel>に格納する任意のプロパティを文字列で指定する. 複数の<application_note>の中から特定のデータを判別・抽出できるような, 手がかりとなる記載を添えておくことが推奨される.	

<application_note>の XML のサンプルは図 12 を参照.

5.5. <reference>

親要素: <voxel>, リンク: -

<reference>は、外部の fav ファイルを指定することで、任意のボクセル集合を 1 ボクセルとして使用することができる。読み込む側の fav ファイルを親、読み込まれる側の fav ファイルを子と呼ぶ時、子の fav ファイルを親の 1 ボクセルとして指定する際は、以下の関係性を保つ必要がある。

親の fav ファイルの <unit.x> = 子の fav ファイルの<unit.x> × <dimension.x>

親の fav ファイルの <unit.y> = 子の fav ファイルの<unit.y> × <dimension.y>

親の fav ファイルの <unit.z> = 子の fav ファイルの<unit.z> × <dimension.z>

図 14 は、外部の fav ファイルを参照したボクセルの定義の例である。実際にモデルデータを定義している「Bunny.fav」では、一辺が 1mm のボクセルの 10×10×10 個の集合である「Voxel01.fav」を 1 ボクセルとして使用している。さらに「Voxel01.fav」では、一辺が 0.01mm のボクセルの 100×100×100 個の集合である「SubVoxel01.fav」を 1 ボクセルとして使用している。

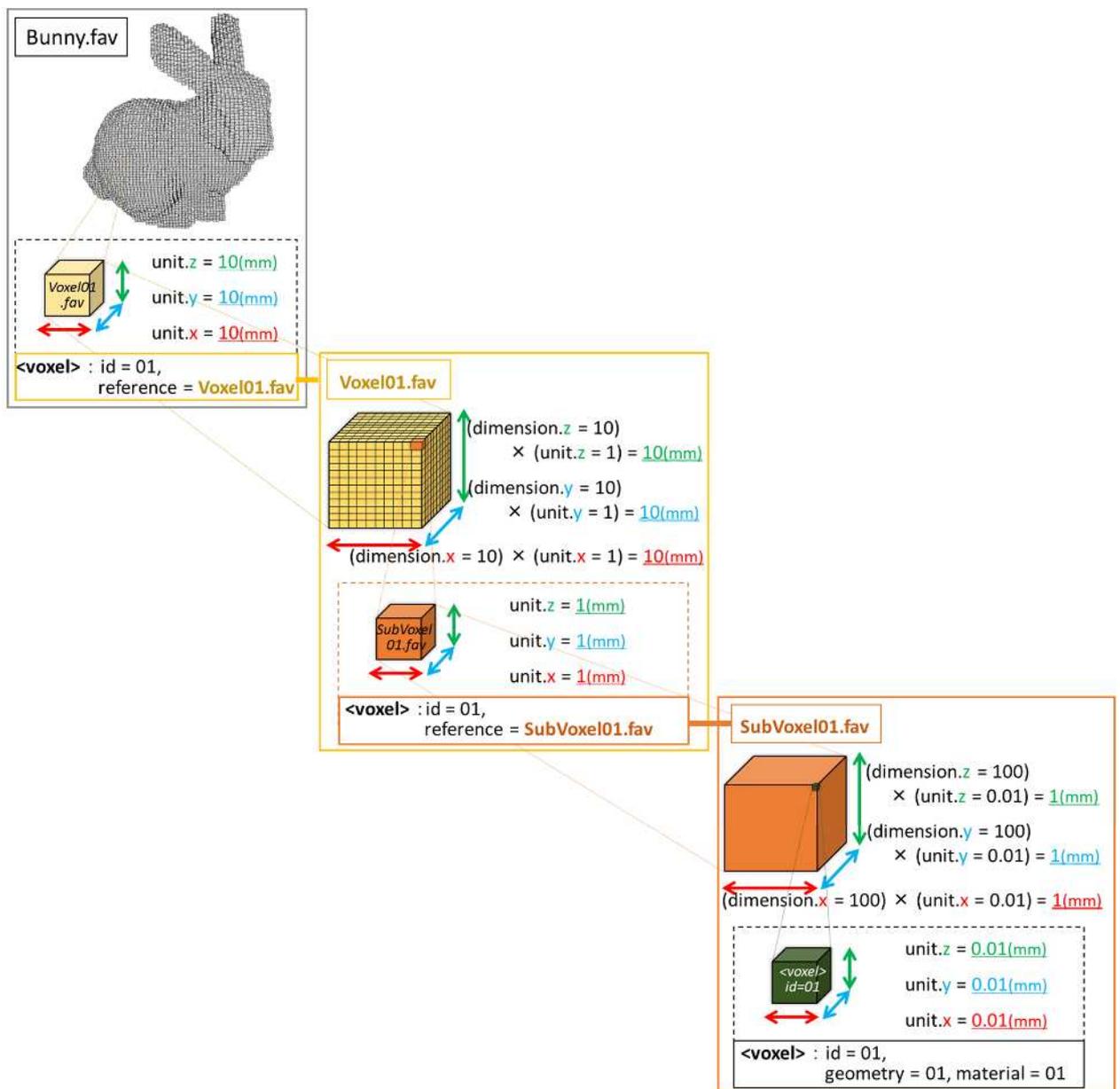


図 14 : 外部 fav ファイルを参照した voxel の定義例 1

図 15 は、外部の fav ファイルを参照したボクセルを複数種類定義する例である。

実際にモデルデータを定義している「Bunny.fav」では、x 辺が 1mm、y 辺が 2.5mm、z 辺が 0.5mm のボクセルの $10 \times 4 \times 20$ 個の集合である「Voxel02.fav」を 1 ボクセルとして使用している。

さらに、「Voxel02.fav」では、それぞれ以下のように 3 種類のボクセルを定義している。

- `<voxel> id=01` … x 辺が 0.05mm、y 辺が 0.05mm、z 辺が 0.01mm のボクセルの $20 \times 50 \times 50$ 個の集合である外部の「SubVoxel_A.fav」を 1 ボクセルとして使用
- `<voxel> id=02` … x 辺が 0.01mm、y 辺が 0.01mm、z 辺が 0.01mm のボクセルの $100 \times 250 \times 50$ 個の集合である外部の「SubVoxel_B.fav」を 1 ボクセルとして使用
- `<voxel> id=03` … x 辺が 1mm、y 辺が 2.5mm、z 辺が 0.5mm のボクセルを「Voxel02.fav」の内部で定義して使用

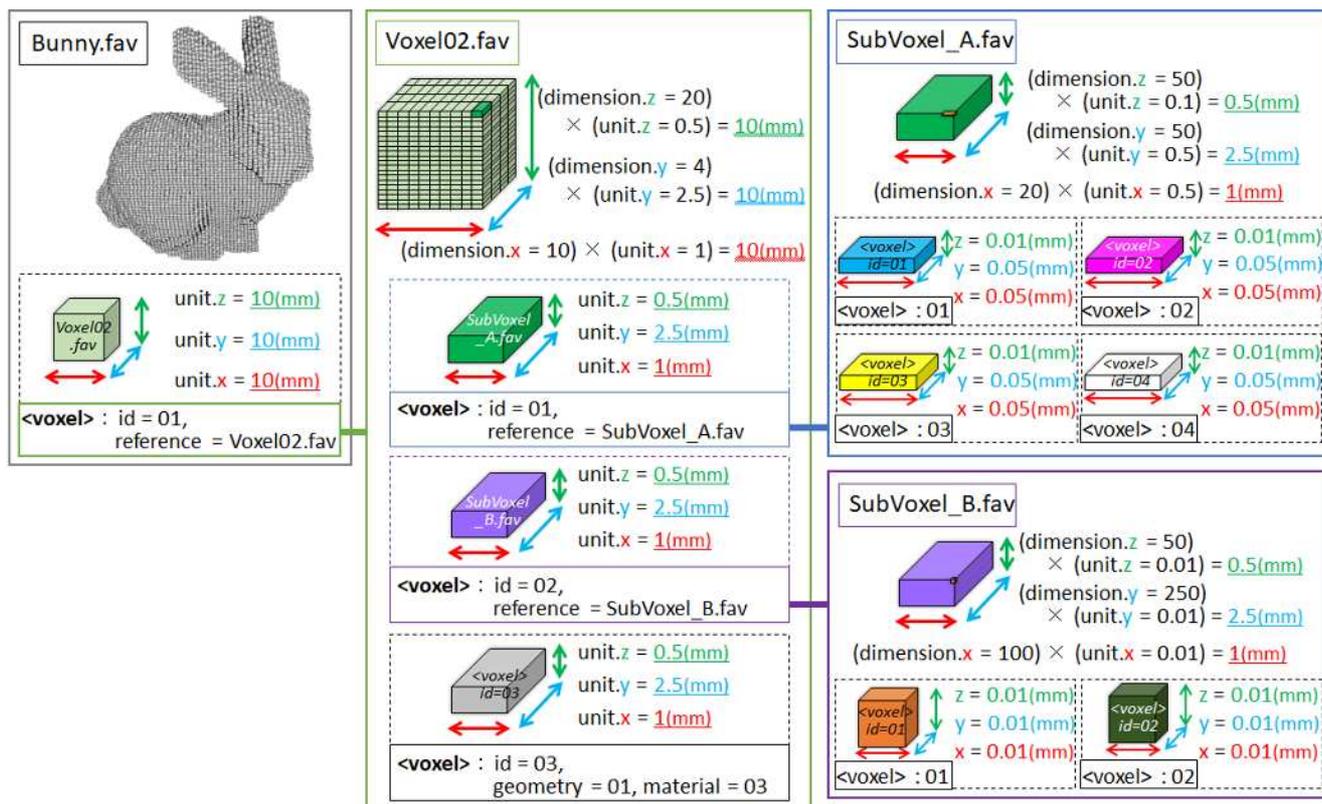


図 15 : 外部 fav ファイルを参照した voxel の定義例 2

`<reference>` は以下の要素を持つ。

表 22: `<reference>` の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
reference	—	CDATA	1 ボクセルとして扱うボクセル集合を定義する fav ファイルパス [—]	1 ボクセルとして扱う外部の fav ファイルまでの相対パスを文字列で指定する。	

`<reference>` の XML のサンプルは図 12 を参照。

6. <object>

親要素: <fav>, リンク: <voxel>, <grid>, <structure>

<object>は, FAV フォーマットにより定義される実際の 3D モデルデータである.

3D モデルデータを格納するための<grid>が定義され, その中に 3D モデルデータの構造である<structure>が記載される. 3D モデルデータは, 形状を表す<voxel_map>, 色情報を表す<color_map>, 接合強度情報を表す<link_map>に分けて記載される. また, ユーザー独自の情報<user_defined_map>を<grid>のセルにマッピングすることができる.

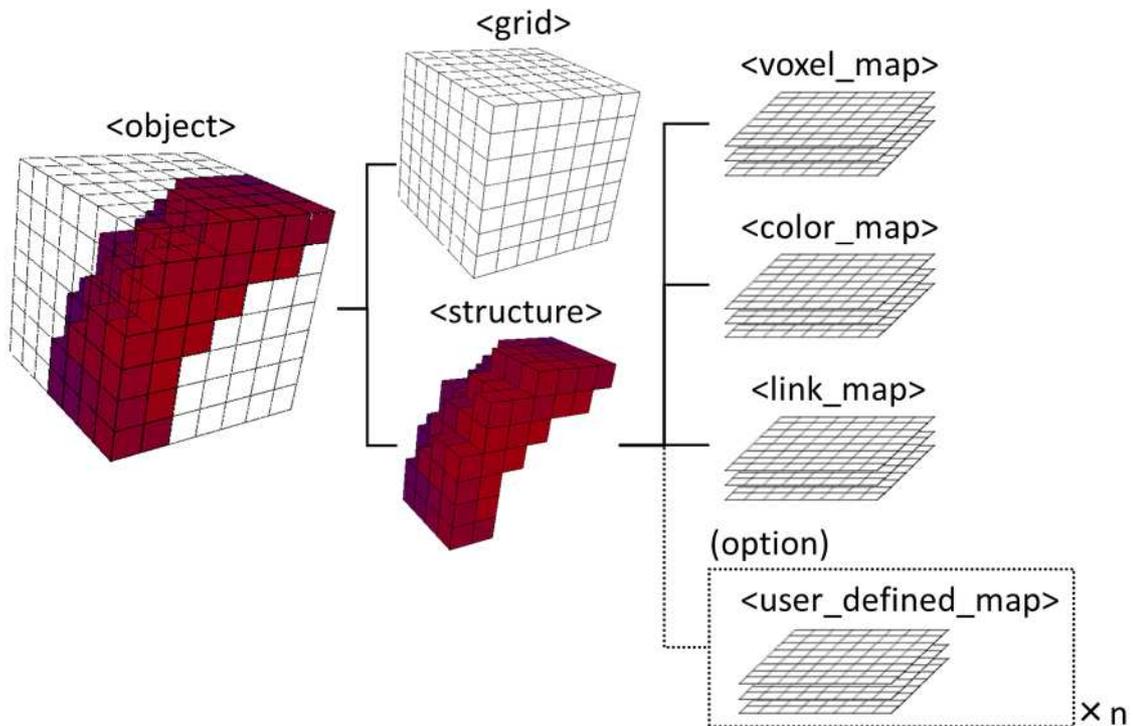


図 16 : Grid と Structure の関係

1 つの FAV フォーマット内で, 複数の<object>を定義することができる. その場合, 各<object>ごとに<grid>と<structure>を定義する.

<object>は以下の属性を持つ.

表 23: <object>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
object	id	positiveInteger	オブジェクトの ID [1*] ※以降, 1 ずつ加算	<object>を一意に示すための ID を設定する. 他の<object>の(id)と重複してはならない.	必須
	name	String	定義するオブジェクトの名称 [Object001~]	<object>の名前を設定する. 他の<object>の(name)と重複しない方が良い.	

<object>以下の階層には, 以下の要素を記載しなければならない.

- <grid>
- <structure>

また, <object>以下の階層に, 以下の要素を記載してもよい. 詳細は<metadata>節で説明する.

- <metadata>

以下に、<object>以下の階層の XML のサンプルを示す。

```

(例)
<object id="1" name="SampleObject">
  <grid>
    <origin>
      <x>28.5</x>
      <y>-30</y>
      <z>0</z>
    </origin>
    <unit>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </unit>
    <dimension>
      <x>7</x>
      <y>7</y>
      <z>7</z>
    </dimension>
  </grid>
  <structure>
    <voxel_map compression="none" bit_per_voxel="8">
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
    </voxel_map>
    <color_map compression="none" color_mode="RGB">
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
    </color_map>
    <link_map bit_per_link="8" compression="none" neighbors="6">
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
      <layer><![CDATA[.....]]</layer>
    </link_map>
    <user_defined_map value_type="byte" compression="none">
      <reference><![CDATA[StressHeatmap.favmapx]]</reference>
      <metadata>
        <id>fa23e6c1-e52e-4591-b354-e4cfa382571a</id>
        <title><![CDATA[Stress Heatmap]]</title>
        <author><![CDATA[Fuji Xerox & Keio SFC]]</author>
        <license><![CDATA[CC BY]]</license>
      </metadata>
    </user_defined_map>
  </structure>
</object>

```

```

    <note><![CDATA[This file is FAVMAP format on ver1.1.]]></note>
  </metadata>
</user_defined_map>
<user_defined_map value_type="float" compression="none">
  <reference><![CDATA[Thermography.favmapx]]></reference>
  <metadata>
    <id>84343e14-ad81-49e9-bf1e-884dcfa80c34</id>
    <title><![CDATA[Thermography sensing data]]></title>
    <author><![CDATA[Fuji Xerox & Keio SFC]]></author>
    <license><![CDATA[CC BY]]></license>
  </metadata>
</user_defined_map>
</structure>
</object>

```

図 17 : <object>以下の階層の XML サンプル

6.1. <grid>

親要素: <object>, リンク: <voxel>, <shape>, <scale>

<grid>は、FAV フォーマットにより定義される実際の 3D モデルデータを定義するための、均等なセルで区切られた空間に関する定義を行う。<grid>に指定した定義に基づき、3D モデルデータの構造の定義を行う。1 つの FAV フォーマット内で、複数の<object>を定義する場合、<object>ごとに<grid>を定義する。

<grid>以下の階層には、以下の要素を記載する。

- <origin>
- <unit>
- <dimension>

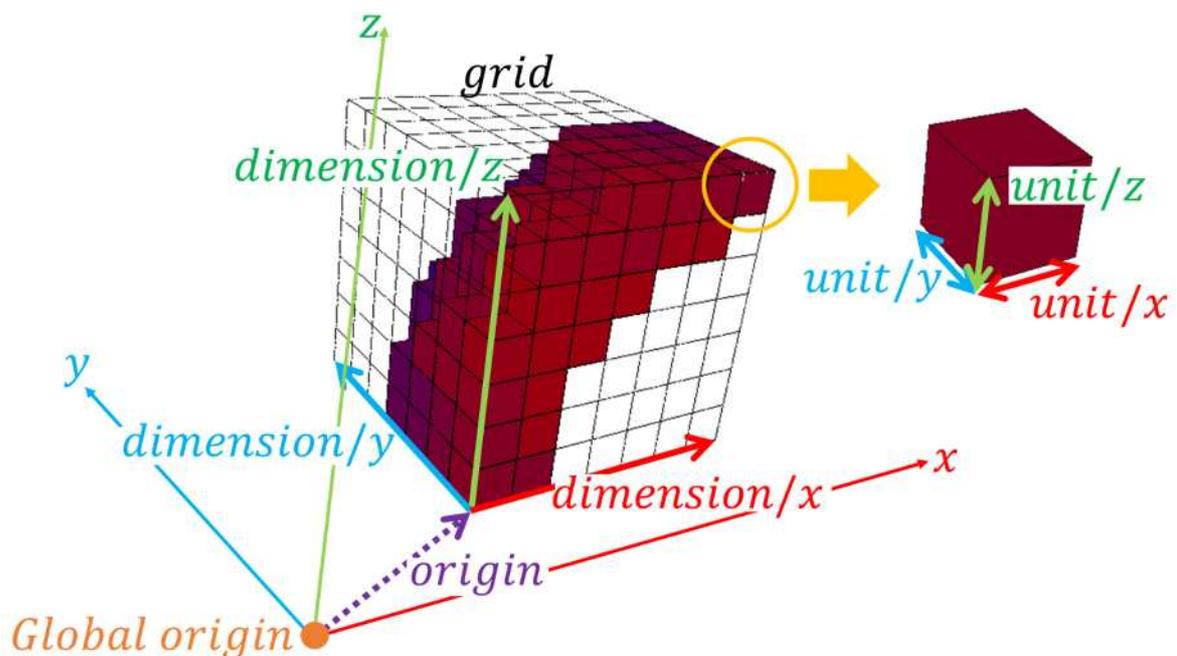


図 18 : <grid>で管理される各要素の意味

6.1.1. <origin>

親要素: <grid>, リンク: -

<origin>は、ワールド空間の原点から<grid>の原点までのオフセット値である。ワールド空間内における、この<object>の位置を指定する。

<origin>は以下の要素を持つ。

表 24: <origin>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
x	—	Double	X 方向の原点 までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの X 軸上のオフセット値を指定する。	
y	—	Double	Y 方向の原点 までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの Y 軸上のオフセット値を指定する。	
z	—	Double	Z 方向の原点 までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの Z 軸上のオフセット値を指定する。	

<origin>の XML のサンプルは図 17 を参照。

6.1.2. <unit>

親要素: <grid>, リンク: <voxel>, <geometry>, <shape>

<unit>は、<grid>のセル 1 つ分の大きさである。X, Y, Z の 3 辺をそれぞれ指定できる。

<unit>から、1mm が何個のボクセルで構成されるか(vpm = voxel per mm)を、下記のように算出できる。

$$1 \text{ (voxel)} / \text{unit (mm)} = \text{vpm}$$

(例) unit = (1, 1, 1) の場合, x = 1 vpm, y = 1 vpm, z = 1 vpm
 unit = (10, 10, 5) の場合, x = 0.1 vpm, y = 0.1 vpm, z = 0.2 vpm
 unit = (0.01, 0.002, 0.05) の場合, x = 100 vpm, y = 500 vpm, z = 20 vpm

<unit>は以下の要素を持つ。

表 25: <unit>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
x	—	Double	X 方向のセル サイズ [1.0]	<grid>のセル 1 つ分の X 方向の大きさを指定する。0 以下を指定してはならない。	
y	—	Double	Y 方向のセル サイズ [1.0]	<grid>のセル 1 つ分の Y 方向の大きさを指定する。0 以下を指定してはならない。	
z	—	Double	Z 方向のセル サイズ [1.0]	<grid>のセル 1 つ分の Z 方向の大きさを指定する。0 以下を指定してはならない。	

<unit>の XML のサンプルは図 17 を参照。

6.1.3. <dimension>

親要素: <grid>, リンク: <voxel>, <unit>

<dimension>は、<grid>全体の大きさである。ボクセルを並べることができる個数で指定する。つまり、グローバル座標空間上の実際の<grid>のサイズは<unit> × <dimension>で求めることができる。

<dimension>は以下の要素を持つ。

表 26: <dimension>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
x	—	positive Integer	X 方向のセル数 [—]	<grid>の X 方向にボクセルを並べることができる個数を指定する。	必須
y	—	positive Integer	Y 方向のセル数 [—]	<grid>の Y 方向にボクセルを並べることができる個数を指定する。	必須
z	—	positive Integer	Z 方向のセル数 [—]	<grid>の Z 方向にボクセルを並べることができる個数を指定する。	必須

<dimension>の XML のサンプルは図 17 を参照。

6.2. <structure>

親要素: <object>, リンク: <voxel_map>, <color_map>, <link_map>, <user_defined_map>

<structure>では、FAV フォーマットにより定義される 3D モデルデータの構造を定義する。<grid>で定義された 3 次元的な grid 内に、<voxel>で定義されたボクセルを立体的に配置していくことで、3D モデルデータの形状を定義する。配置したボクセルの色や材料、接合強度などの属性を各種 map で定義する。

<structure>以下の階層には、以下の要素が記載される

- <voxel_map>
- <color_map>
- <link_map>
- <user_defined_map>

<user_defined_map>を定義することで、ユーザー独自の属性を<grid>のセルにマッピングすることができる。<user_defined_map>は、複数定義することができる。

<structure>以下の階層の各要素について、以下のような 3D モデルデータを定義する場合を例に解説する。

<structure>の XML のサンプルは図 30 を参照。

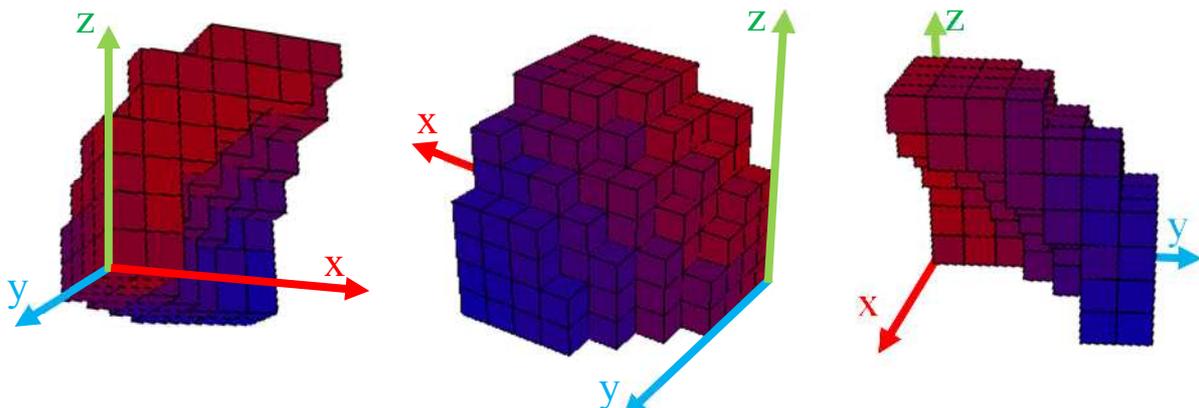


図 19 : FAV フォーマットで定義される 3D モデルデータの例

6.2.1. <voxel_map>

親要素: <structure>, リンク: <grid>, <voxel>

<voxel_map>では, FAV フォーマットにより定義される 3D モデルデータの形状を定義する. <grid>で定義された 3 次元的な grid の XY 平面 1 層ごとに, 層内の各 grid 位置におけるボクセルの有無を定義する. 層ごとに定義された voxel map を, 3D モデルデータの高さ分積み上げることで, 3D モデルデータ全体の形状を定義する.

<voxel_map>は, 以下の属性を持つ.

表 27: <voxel_map>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
voxel_map	bit_per_voxel	positive integer	4 / 8 / 16 [—]	ボクセル 1 個が何 bit(16 進文字列何文字)で表現されるかを 4 / 8 / 16 のいずれかで指定する. (それぞれ 16 進文字列 1 文字 / 2 文字 / 4 文字)	必須
	compression	String	none / base64 / zlib / runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する.	

<voxel_map>以下の階層には, <dimension.z>分の<layer>が定義される. 3D モデルデータの接地面, つまり高さ方向の一番低い層から順番に<layer>が格納される.

1 つの<layer>には, XY 平面 1 層分の voxel map が記載される. Voxel map に<voxel>の(id)を記載した場合, Voxel map で表される grid 位置に(id)に相当するボクセルが存在することを示す. Voxel map で表される grid 位置にボクセルが存在しない場合は 0 を記載する.

<voxel_map.layer>は, 以下のデータを持つ.

表 28: <voxel_map.layer>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
layer	—	CDATA	ボクセルの ID を列挙した文字列 [—]	<voxel>の(id)を(bit_per_voxel)で指定したフォーマットで列挙する. <dimension.x> × <dimension.y>分の(id)が 1 行に連結されて記載される. <voxel_map>の(compression)に none 以外を指定した場合, <layer>のデータは指定の圧縮方法で圧縮されて格納される.	

図 19 の 3D モデルデータの 1 層目を記載した voxel map は以下ようになる.

(voxel map のセルの塗りつぶしは可視性を考慮したものであり, データとしての意味はない)

- ① <grid>の<dimension.x>1 列分の各位置におけるボクセルの有無を, <voxel>の(id)で列挙する. ボクセルがない場合は 0 を記載する.

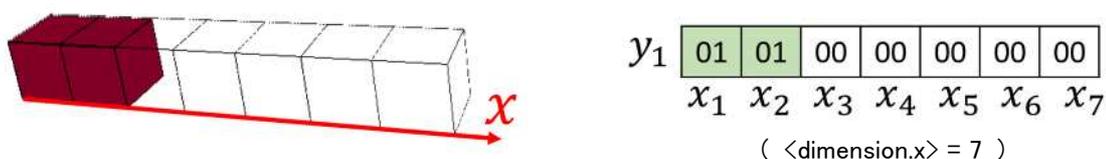
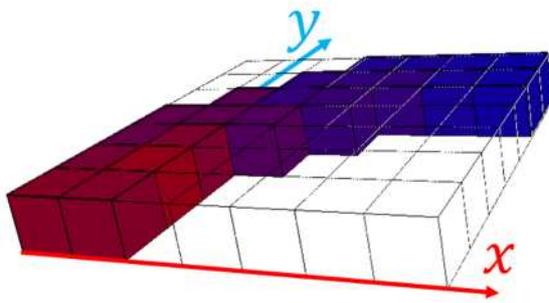


図 20 : <dimension.x>1 列分の<voxel_map>

- ② <grid>の<dimension.x>分列挙した voxel map を、<dimension.y>分列挙する。



y ₇	00	00	00	02	02	02	02
y ₆	00	00	01	02	02	02	02
y ₅	00	01	01	01	00	00	00
y ₄	01	01	01	00	00	00	00
y ₃	01	01	00	00	00	00	00
y ₂	01	01	00	00	00	00	00
y ₁	01	01	00	00	00	00	00
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇

(<dimension.x> = 7, <dimension.y> = 7)

図 21 : <dimension.x> × <dimension.y> 1 層分の <voxel_map>

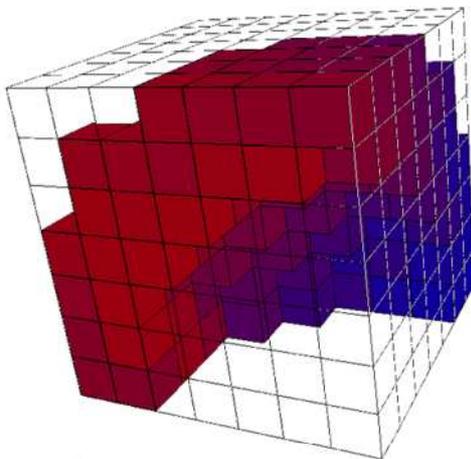
- ③ <grid>の<dimension.x> × <dimension.y>分の voxel map を 1 行に連結したものが、1 層目の <layer>のデータとなる。

y ₁	01	01	00	00	00	00	00	y ₂	01	01	00	00	...	y ₇	02	02	00	00	00	02	02	02	02
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₁	x ₂	x ₃	x ₄	...	x ₆	x ₇	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇		

<layer>0101000000000001010000...020200000002020202</layer>

図 22 : 1 層分の <voxel_map> を 1 列に連結した <layer> のデータ

- ④ <grid>の<dimension.x> × <dimension.y> を 1 行で列挙した 1 層分の <layer> を、<dimension.z> 分定義することにより、3D モデルデータ全体の形状を定義する。



<layer>0101000000000001010000...020200000002020202</layer>

...

<layer>0000000101010100000001010101000000...000000</layer>

Z ₁	y ₁	01	01	00	00	00	00	00	01	01	00	00	...	y ₇	02	02	00	00	00	02	02	02	02
		x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₁	x ₂	x ₃	x ₄	...	x ₆	x ₇	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	
	⋮	(<dimension.z> = 7)																					
Z ₇	y ₁	00	00	00	01	01	01	01	00	00	00	01	01	01	01	00	00	00	...	y ₃	00	00	00
		x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₁	x ₂	x ₃	...	x ₅	x ₆	x ₇	

図 23 : 3D モデルデータ全体の <voxel_map>

6.2.2. <color_map>

親要素: <structure>, リンク: <grid>, <voxel>, <display>

<color_map>では, FAV フォーマットにより定義される 3D モデルデータの色情報を定義する.

<voxel_map>で列挙された各ボクセルに対し, カラー情報を指定する.

<color_map>は, 以下の属性を持つ.

表 29: <color_map>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
color_map	color_mode	String	GrayScale / GrayScale16 / RGB / RGBA / CMYK [—]	ボクセルに定義される色情報のフォーマットを, GrayScale / GrayScale16 / RGB / RGBA / CMYK のいずれかで指定する.	必須
	compression	String	none / base64 / zlib / runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する.	

(color_mode)の各色情報フォーマットには, 以下の物がある.

- GrayScale ... 白～黒までのGrayScaleが1チャンネルにつき256階調(1byte)で表現される. 1ボクセル分の色情報は1byte(16進数2文字)で記載される.
- GrayScale16 ... 白～黒までのGrayScaleが1チャンネルにつき65,536階調(2byte)で表現される. 1ボクセル分の色情報は2byte(16進数4文字)で記載される.
- RGB ... Red, Green, Blueの3チャンネルがそれぞれ256階調(1byte)で表現される. 1ボクセル分の色情報は3byte(16進数6文字)で記載される.
- RGBA ... Red, Green, Blue, Alphaの4チャンネルがそれぞれ256階調(1byte)で表現される. 1ボクセル分の色情報は4byte(16進数8文字)で記載される.
- CMYK ... Cyan, Magenta, Yellow, KeyPlateの4チャンネルがそれぞれ256階調(1byte)で表現される. 1ボクセル分の色情報は4byte(16進数8文字)で記載される.

<color_map>以下の階層には, <dimension.z>分の<layer>が定義される. 1つの<layer>には, XY平面1層分のcolor mapが記載される. color mapには, <voxel_map>に列挙したボクセルそれぞれに対し, (color_mode)に指定したフォーマットで色情報を列挙する. <voxel_map>でボクセルが存在しないことを表す0が指定された場合, <color_map>上での色情報の指定は省略され, 左詰めにして列挙する.

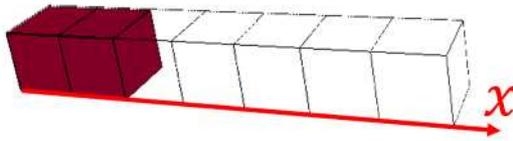
<color_map.layer>は, 以下のデータを持つ.

表 30: <color_map.layer>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
layer	—	CDATA	3Dモデルの色情報を表す16進数文字列 [—]	<voxel_map>に列挙したボクセルの色情報を, (color_mode)に指定したフォーマットで列挙する. <color_map>の(compression)にnone以外を指定した場合, <layer>のデータは指定の圧縮方法で圧縮されて格納される.	1以上

図 19 の 3D モデルデータの 1 層目を記載した color map は以下ようになる。
 (voxel map のセルの塗りつぶしは可視性を考慮したものであり、データとしての意味はない)

- ① <grid>の<dimension.x>1 列分の各位置におけるボクセルの色情報を、(color_mode)に指定したフォーマットで列挙する。ボクセルが存在しないことを示す 0 が指定されていた場合、省略する。



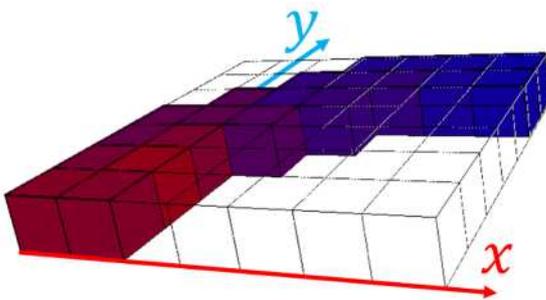
y ₁	01	01	00	00	00	00	00
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇

voxel_y₁x₁ = RGB(131, 0, 37) → 830025
 voxel_y₁x₂ = RGB(129, 0, 39) → 810027

以降は 0 のため省略される。
 <dimension.x>1 列分の color map は 830025810027 となる。

図 24 : <dimension.x>1 列分の<color_map>

- ② <grid>の<dimension.x>分列挙した color map を、<dimension.y>分列挙する。<dimension.x> × <dimension.y>分の color map を 1 行に連結したものが、1 層分の<layer>のデータとなる。



y ₇	00	00	00	02	02	02	02
y ₆	00	00	01	02	02	02	02
y ₅	00	01	01	01	00	00	00
y ₄	01	01	01	00	00	00	00
y ₃	01	01	00	00	00	00	00
y ₂	01	01	00	00	00	00	00
y ₁	01	01	00	00	00	00	00
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇

1 層分の layer に存在する 21 個の voxel に定義された color map のデータ例
 <Layer>8300258100277600329100176400457c002d5e004a5c004c5000595600523300753700713000782f007a3
 100771800900f00991f00891c008c1300960c009c</Layer>

図 25 : <dimension.x> × <dimension.y>1 層分の<color_map>

- ③ <dimension.x> × <dimension.y>分の color map を 1 行に連結した 1 層分の<layer>を、<dimension.z>分定義することにより、3D モデルデータ全体の色情報を定義する。

6.2.3. <link_map>

親要素: <structure>, リンク: <grid>, <voxel>

<link_map>では、FAV フォーマットにより定義される 3D モデルデータのリンク情報を定義する。リンク情報とは、ボクセル間の関係性の強度を表す値である。例えば、ボクセル間の接合強度などを表現する。これらのリンク情報は、より精度の高い構造解析や 3D プリンターのツールパスの生成などに用いることができる。<voxel_map>で列挙された各ボクセルに対し、リンク情報を指定する。

<link_map>は、以下の属性を持つ。

表 31: <link_map>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
link_map	bit_per_link	positive Integer	4 / 8 / 16 [—]	リンク情報 1 個が何 bit(16 進文字列何文字)で表現されるかを 4 / 8 / 16 のいずれかで指定する。(それぞれ 16 進文字列 1 文字 / 2 文字 / 4 文字)	必須
	neighbors	positive Integer	6 / 18 / 26 [—]	対象ボクセルの周囲何近傍までリンク情報を保持するかを 6 / 18 / 26 のいずれかで指定する。	必須
	compression	String	none / base64 / zlib / runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する。	

対象ボクセルの周囲何近傍までリンク情報を保持するかを指定する (**neighbors**) は以下の 3 種類がある(エラー! 参照元が見つかりません。).

- 6 近傍 ... 該当ボクセルと面で接する 6 個のボクセルとの間のリンク情報を表現する。
- 18 近傍 ... 該当ボクセルと辺で接する 18 個のボクセルとの間のリンク情報を表現する。
- 26 近傍 ... 該当ボクセルと点で接する 26 個のボクセルとの間のリンク情報を表現する。

1 個の Voxel が管理するリンク情報は、(**bit_per_link**)分の 16 進文字数 × (**neighbors**) 個分で表される。下の(例 1)、(例 2)は<link_map>の中から 1 個分の Voxel のリンク情報の記載を抜き出した例である。

```
(例 1) <link_map bit_per_link=4, neighbors=6>
      <layer>...004af0...</layer>
</link_map>
```

16 進数 1 文字 × 6 近傍 = 1 ボクセル分のリンク情報は 16 進数 6 文字で記載される

```
(例 2) <link_map bit_per_link=16, neighbors=26>
      <layer>...
          000000000000004e01a900000000000000000000f
          0cbe0000100d0000005000000000000000000000
          00200ff60000000000000000...
      </layer>
</link_map>
```

16 進数 4 文字 × 26 近傍 = 1 ボクセル分のリンク情報は 16 進数 104 文字で記載される

<link_map>以下の階層には、<dimension.z>分の<layer>が定義される。1 つの<layer>には、XY 平面 1 層分のリンク情報が記載される。1 つの<layer>には、<voxel_map>に列挙したボクセルそれぞれに対し、(**neighbors**)に指定したフォーマットでリンク情報を列挙する。<voxel_map>でボクセルが存在しないことを表す 0 が指定された場合、

<link_map>上でのリンク情報の指定は省略され、左詰めにして列挙する。

隣接するボクセルが存在しない場合、リンク情報に 0 を指定し、必ず(neighbors)で指定されたフォーマットで定められた byte 数分のリンク情報を列挙しなければならない。

<link_map>上でのリンク情報の記載順序は、以下の図で示す通り、z,y,x の座標値の小さい順とする(エラー! 参照元が見つかりません。). 赤色のボクセルは対象ボクセル、数字は記載順序を意味する。

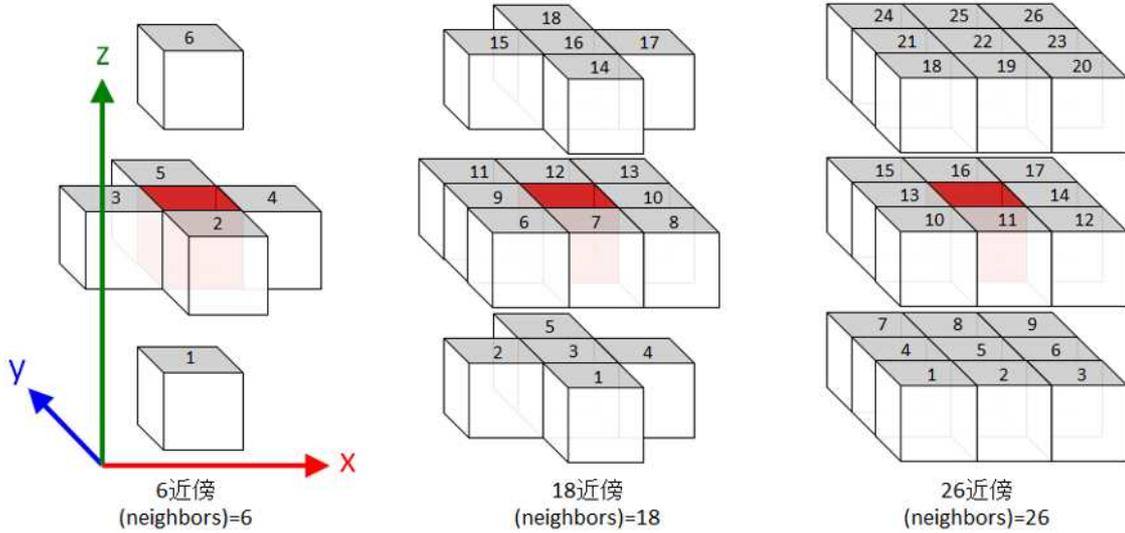


図 26: 各近傍(neighbors)フォーマットの記載順序

<link_map.layer>は、以下のデータを持つ。

表 32: <link_map.layer>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
layer	—	CDATA	3D モデルのリンク情報を表す 16 進数文字列 [—]	<voxel_map>に列挙したボクセルのリンク情報を、(neighbors)に指定したフォーマットで、エラー! 参照元が見つかりません。の順番に列挙する。 <link_map> の (compression)に none 以外を指定した場合、<layer>のデータは指定の圧縮方法で圧縮されて格納される。	1 以上

図 19 の 3D モデルデータのうち、特定の個所の Voxel に対して定義されるリンクデータは以下ようになる。

(neighbors)に 6 を指定したとき, x 軸方向, y 軸方向, z 軸方向のリンク情報をそれぞれ 100, 200, 255 とした場合, 図 27 の voxel1, voxel2 に示される個所のリンク情報は以下のように記述する.

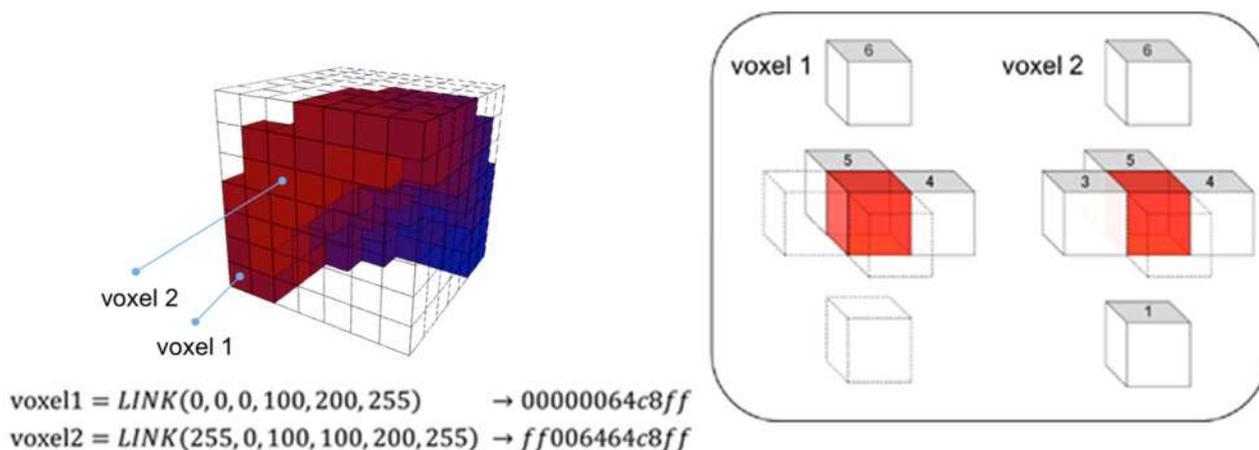
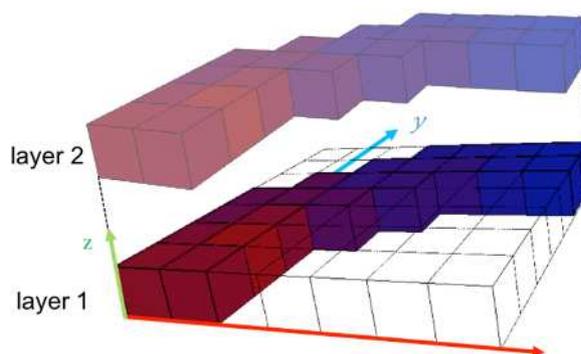


図 27 : 隣接ボクセルに対するリンク情報

以上のように定義されるボクセル 1 つ分のリンク情報を, $\langle \text{dimension.x} \rangle \times \langle \text{dimension.y} \rangle$ 分列挙することで, 1 層分のリンク情報のマップを定義する. それを 1 行に連結したものが, 1 層分の $\langle \text{layer} \rangle$ のデータとなる. $\langle \text{layer} \rangle$ を $\langle \text{dimension.z} \rangle$ 分定義することにより, 3D モデルデータ全体のリンク情報を定義する.



layer1 に存在する 21 個のボクセルに定義された link map のデータ例

```
<layer>00000064c8ff0000000c8ff00000064c8ff0000000c8ff00c80064c8ff00c80000c8ff00c8006400ff00c80064c8ff00006400c8ff00c8006400ff00c86464c8ff00006400c8ff00c8006400ff00c86464c8ff00006464c8ff00006464c8ff0006400c8ff00c8006400ff00c8646400ff00c8646400ff00c8640000ff</layer>
```

図 28 : $\langle \text{dimension.x} \rangle \times \langle \text{dimension.y} \rangle$ 層分の $\langle \text{link_map} \rangle$

6.2.4. <user_defined_map>

親要素: <structure>, リンク: <.favmap>, <.favmap>

<user_defined_map>では, FAV フォーマットにより定義される 3D モデルデータを構成する各<grid>のセルに対し, ユーザー独自の属性を付与するためのマップを定義する. ここでは, ユーザー定義属性の付与対象となるファイルを, 親ファイルと呼ぶ.

ユーザー定義属性は, 親ファイルの<grid>のセル 1 つに対し, 1 つの属性値を対応付ける. ユーザー定義属性のマップを複数定義することで, 親ファイルの<grid>のセル 1 つに対し, 複数の属性を対応付けることができる.

<user_defined_map>は, <reference>を定義することで, 外部ファイルからユーザー定義属性を読み込む. 外部ファイルは, XML 形式(拡張子「.favmapx」), または Binary 形式(拡張子「.favmap」)でユーザー定義属性を記述する. ユーザー定義属性ファイルの詳細な仕様は次節で説明する.

<user_defined_map>は, 以下の属性を持つ.

表 33: <user_defined_map>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
user_defined_map	value_type	String	byte / short / ushort / int / uint / float / double [byte]	外部ファイルに格納されているユーザー定義属性の, ボクセル 1 個分の属性値の型を指定する.	必須
	compression	String	none / base64 / zlib / runlength [none]	外部ファイルに格納されている<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する. (XML 形式の「.favmapx」ファイル時のみ有効)	

<user_defined_map>以下の階層には, 以下の要素が記載される.

- <reference>
- <metadata>

<user_defined_map>を定義したユーザーだけでなく, データを利用する別のユーザーにも, 定義されている属性の種類や読み込む必要性が判断できるように, <metadata>を定義することが推奨される.

表 34: <user_defined_map.reference>, <user_defined_map.metadata>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
reference	—	CDATA	ユーザー定義属性を格納する外部ファイルパス [—]	ユーザー定義属性が格納されている外部の.favmap/.favmapx ファイルまでの相対パスを文字列で指定する.	必須
metadata	—	<metadata>参照		ユーザー定義属性を特定するための ID, 属性の名称 (Title), 発行者 (Author), 利用許諾 (License), 属性の説明 (note), 等を記載する.	

6.2.4.1. ユーザー定義属性ファイル – XML 形式 (.favmapx ファイル)

親要素: `<user_defined_map>`, リンク: `<grid>`, `<voxel>`

`<user_defined_map>`の`<reference>`で指定される, ユーザー定義属性を格納する外部の「.favmapx」ファイルの仕様を説明する. ここでは, 「.favmapx」ファイルを参照する側の「.fav」ファイルを, 親ファイルと呼ぶ.

XML 形式でユーザー定義属性を格納する場合, 以下のように記述される.

```
(例)
<?xml version="1.0" encoding="utf-8"?>
<fav version="1.1">
  <user_defined_map>
    <layer><![CDATA[.....]]</layer>
    <layer><![CDATA[.....]]</layer>
    <layer><![CDATA[.....]]</layer>
    <layer><![CDATA[.....]]</layer>
  </user_defined_map>
</fav>
```

図 29 : `<geometry>`以下の階層の XML サンプル

`<user_defined_map>`以下の階層には, 親ファイルの`<dimension.z>`分の`<layer>`が定義される. 1 つの`<layer>`には, XY 平面 1 層分のユーザー定義属性のマップが記載される. 1 つの`<layer>`には, 親ファイルの(`value_type`)で指定された型の属性値が, 親ファイルの`<dimension.x>` × `<dimension.y>`の数, 列挙される.

親ファイルの(`value_type`)に指定されたそれぞれの型は, 以下のようにデータが記載される.

- `byte` ... 1 セルのユーザー定義属性値は 0 ~ 255 の範囲で表される.
1 セルのユーザー定義属性値は 1byte (16 進数 2 文字) で記載される.
- `short` ... 1 セルのユーザー定義属性値は -32,768 ~ 32,767 の範囲で表される.
1 セルのユーザー定義属性値は 2byte (16 進数 4 文字) で記載される.
- `ushort` ... 1 セルのユーザー定義属性値は 0 ~ 65,535 の範囲で表される.
1 セルのユーザー定義属性値は 2byte (16 進数 4 文字) で記載される.
- `int` ... 1 セルのユーザー定義属性値は -2,147,483,648 ~ 2,147,483,647 の範囲で表される.
1 セルのユーザー定義属性値は 4byte (16 進数 8 文字) で記載される.
- `uint` ... 1 セルのユーザー定義属性値は 0 ~ 4,294,967,295 の範囲で表される.
1 セルのユーザー定義属性値は 4byte (16 進数 8 文字) で記載される.
- `float` ... 1 セルのユーザー定義属性値は $\pm 10^{-37} \sim \pm 10^{38}$ の範囲で表される.
1 セルのユーザー定義属性値は 4byte (16 進数 8 文字) で記載される.
- `double` ... 1 セルのユーザー定義属性値は $\pm 10^{-307} \sim \pm 10^{308}$ の範囲で表される.
1 セルのユーザー定義属性値は 8byte (16 進数 16 文字) で記載される.

親ファイルの`<voxel_map>`でボクセルが存在しないことを表す 0 が指定された場合でも, `<user_defined_map>`へ属性情報を列挙する. 属性情報が不要な場合は 0 を指定する.

「.favmapx」ファイルは, 親ファイルと以下の関係性を保つ必要がある。

- ・ 親ファイルの`<voxel_map>`以下の`<layer>`数と, `<user_defined_map>`以下の`<layer>`数が等しいこと
- ・ 親ファイルの`<voxel_map>`の`<layer>`に列挙するデータ数と, `<user_defined_map>`の`<layer>`に列挙するデータ数が等しいこと (データ数が親ファイルの `<dimension.x>` × `<dimension.y>` となること)

`<user_defined_map>`以下に, 複数の`<layer>`が格納される. `<layer>`は, 以下のデータを持つ.

表 35: <user_defined_map.layer>の説明

要素ラベル	属性ラベル	型	データ [デフォルト値]	説明	条件
layer	—	CDATA	ユーザー定義 属性を列挙した 文字列 [—]	親ファイルの<grid>各セルの属性情報を、 親ファイルの (value_type) に指定したフ ォーマットで列挙する。親ファイルの <user_defined_map> の (compression) に none 以外を指定した場合、<layer> のデータは指定の圧縮方法で圧縮され て格納される。	

6.2.4.2. ユーザー定義属性ファイル – Binary 形式 (.favmap ファイル)

親要素: <user_defined_map>, リンク: <grid>, <voxel>

<user_defined_map>の<reference>で指定される, ユーザー定義属性を格納する外部の「.favmap」ファイルの仕様を説明する. ここでは, 「.favmap」ファイルを参照する側の「.fav」ファイルを, 親ファイルと呼ぶ.

Binary 形式でユーザー定義属性を格納する場合, 親ファイルの(<value_type>)に指定されたそれぞれの型は, 以下のようにデータが格納される. データは, Binary ファイルの先頭から順番に格納される.

- byte ... 1 セルのユーザー定義属性値は 0 ~ 255 の範囲で表される.
1 セルのユーザー定義属性値は 1byte で格納される.
- short ... 1 セルのユーザー定義属性値は -32,768 ~ 32,767 の範囲で表される.
1 セルのユーザー定義属性値は 2byte で格納される.
- ushort ... 1 セルのユーザー定義属性値は 0 ~ 65,535 の範囲で表される.
1 セルのユーザー定義属性値は 2byte で格納される.
- int ... 1 セルのユーザー定義属性値は -2,147,483,648 ~ 2,147,483,647 の範囲で表される.
1 セルのユーザー定義属性値は 4byte で格納される.
- uint ... 1 セルのユーザー定義属性値は 0 ~ 4,294,967,295 の範囲で表される.
1 セルのユーザー定義属性値は 4byte で格納される.
- float ... 1 セルのユーザー定義属性値は $\pm 10^{-37} \sim \pm 10^{38}$ の範囲で表される.
1 セルのユーザー定義属性値は 4byte で格納される.
- double ... 1 セルのユーザー定義属性値は $\pm 10^{-307} \sim \pm 10^{308}$ の範囲で表される.
1 セルのユーザー定義属性値は 8byte で格納される.

Binary ファイルの先頭から, 以下のように属性情報が格納される.

- ① 親ファイルの<grid>の地面に近い層, つまり高さ方向の一番低い層から格納する.
親ファイルの<grid>の<dimension.x>1 列分の属性情報を, Binary ファイルの先頭から順に格納する.
属性情報がない場合は 0 を格納する.
- ② そのまま続けて次の<dimension.x>1 列分の属性情報を格納する.
- ③ ②を親ファイルの<grid>の<dimension.y>分, 続けて格納する. 親ファイルの <dimension.x> × <dimension.y> 分を格納した時点で, XY 平面 1 層分のユーザー定義属性となる.
- ④ そのまま続けて次の層の②-③を繰り返す. 親ファイルの <dimension.x> × <dimension.y> × <dimension.z> 分を格納した時点で, 親ファイルの<grid>全体分のユーザー定義属性となる.

Binary ファイル内にはデータの区切りや<layer>の区切りを表す情報は含まれず, すべてのデータはユーザー定義属性を表す.

「.favmap」ファイルのサイズは, 以下の式によって求められる.

サイズ(byte) = 親ファイルの <dimension.x> × <dimension.y> × <dimension.z> × (value_type)で示す型の byte 数

7. まとめ

本仕様書で説明した FAV フォーマットの全要素・全属性の一覧を記載する。

表 36: FAV フォーマットの全 Element, および Attribute

要素ラベル (左列に行くほど親階層, 右列に行くほど子階層を表す)		属性ラベル	型	データ [デフォルト値]	説明	条件	
—	fav	—	—	—	FAV フォーマットの Root 要素である。 <metadata>, <palette>, <voxel>, <object>を定義する。	必須	
		version	String	FAV フォーマットのバージョン番号 [—]	この FAV ファイルが則しているフォーマットのバージョン番号を文字列で指定する。	必須	
fav / material / object / user_defined_map	metadata	→	—	—	<metadata> には, <id>, <title>, <author>, <license>, <note>を定義する。		
		id	—	String	親要素の ID [—]	<metadata>を定義する親要素を一意に示す ID を文字列で指定する。	必須
		title	—	CDATA	親要素の名前 [—]	<metadata>を定義する親要素を表す名前を文字列で指定する。	必須
		author	—	CDATA	親要素の作者 [—]	<metadata>を定義する親要素の作者情報を文字列で指定する。	必須
		license	—	CDATA	親要素のライセンス情報 [—]	<metadata>を定義する親要素のライセンス情報を文字列で指定する。	必須
		note	—	CDATA	親要素に関するメモ [—]	<metadata>を定義する親要素に関するメモを文字列で指定する。	

fav	palette	→		—	—	—	<palette> には, <geometry>, <material>を定義する.		
				—	—	—	<geometry> には, <shape>, <scale>を定義する.	複数定義可	
		→		id	positive Integer	幾何情報の ID [1*] *以降, 1 ずつ加算	<geometry>を一意に示すための IDを設定する. 他の<geometry>の(id)と重複してはならない.	必須	
				name		String	定義する幾何情報の名称 [—]	<geometry>の名前を設定する. 他の<geometry>の(name)と重複しない方が良い.	
				shape		—	String	cube / sphere / user_defined [cube]	Voxel の形状を指定する. user_defined を指定した場合, <reference>を追加で指定しなければならない.
		reference		—	CDATA	voxel の形状を定義する stl ファイルパス [—]	Voxel の形状を定義する, 外部の stl ファイルまでの相対パスを文字列で指定する.		
		scale		x	—	Double	X 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の X 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	
				y	—	Double	Y 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の Y 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	
				z	—	Double	Z 方向の倍率 [1.0]	<grid>のセルを基準として, <shape>の Z 方向の倍率を指定する. 負の値が設定された場合, 反転する. 0 を指定してはならない.	

fav	palette	material	→		—	—	—	<material>には<material_name>, <product_info>, <standard_name>をいずれか1つ以上定義する.	複数定義可	
			→		id	positive Integer	材料情報の ID [1*] ※以降, 1 ずつ加算	<material>を一意に示すための ID を設定する. 他の<material>の(id)と重複してはならない.	必須	
			→		name	String	材料情報の定義名 [—]	<material>の名前を設定する. 他の<material>の(name)と重複しない方が <u>良い</u> .		
			material_name	→	—	CDATA	材料の名前情報 [—]	なんの材料を使用するかを判別することができる文字列を指定する.	複数定義可	
			product_info	→	—	—	—	以下の要素を持つ.	複数定義可	
				manufacturer	→	—	CDATA	材料メーカー名 [—]	材料のメーカー名を文字列で指定する.	
				product_name	→	—	CDATA	材料の製品名 [—]	材料を表すメーカー指定の商品名, 商品コード, 等を文字列で指定する.	
				url	→	—	CDATA	材料に関する web 上の情報の url [—]	材料に関する web 上の情報へのリンクを url で指定する.	
			standard_name	→	—	CDATA	材料の規格情報 [—]	材料を規格に定義された名前に従って指定する.	複数定義可	
				→	—	—	—	—	—	
→	—	—		—	—	—				

fav	voxel		→	—	—	—	<voxel> には <geometry_info>, <material_info>を記載しなければならない。または、<reference>のみを定義する。<display>, <application_note>を定義することができる。	複数定義可
			id	—	positive Integer	ボクセルの ID [1*] *以降, 1 ずつ加算	<voxel>を一意に示すための ID を 1 以上の整数で設定しなければならない。他の<voxel>の(id)と重複してはならない。	必須
			name	—	String	定義するボクセルの名称 [—]	<voxel> の名前を設定する。他の<geometry>の(name)と重複しない方が良い。	
	geometry_info		→	—	—	—		
			id	—	positive Integer	定義済みの形状情報の ID [—]	<geometry>を参照するための ID を指定する。<geometry>に設定済みの(id)を指定しなければならない。	必須
	material_info		→	—	—	—		複数定義可
			id	—	positive Integer	定義済みの材料情報の ID [—]	<material>を参照するための ID を指定する。<material>に設定済みの(id), もしくは 0 を指定しなければならない。	必須
			ratio	—	Double	複数の材料の複合割合 [—]	<material>を使用する割合を Double 値で指定する。他の<material>の割合と合計で 1.0 になるように指定する必要がある。0 以下を指定してはならない。	
	display		r	—	nonNegativeInteger	ボクセル表示色の赤(Red)成分 [—]	<voxel>の色情報を表す Red 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	
			g	—	nonNegativeInteger	ボクセル表示色の緑(Green)成分 [—]	<voxel>の色情報を表す Green 成分を指定する。0 未満, または 256 以上の値を指定してはならない。	

fav	voxel	display	b	—	nonNegativeInteger	ボクセル表示色の青(Blue)成分 [—]	<voxel>の色情報を表す Blue 成分を指定する. 0 未満, または 256 以上の値を指定してはならない.		
			a	—	nonNegativeInteger	ボクセル表示色の透過度 [255]	<voxel>の色情報を表す Alpha 成分を指定する. 0 未満, または 256 以上の値を指定してはならない.		
		application_note	→	—	CDATA	ボクセルに格納する任意のプロパティ情報 [—]	<voxel>に格納する任意のプロパティを文字列で指定する. 複数の<application_note>の中から特定のデータを判別・抽出できるような, 手がかかりとなる記載を添えておくことが推奨される.		
		reference	—	—	CDATA	1 ボクセルとして扱うボクセル集合を定義する fav ファイルパス [—]	1 ボクセルとして扱う外部の fav ファイルまでの相対パスを文字列で指定する.		
	object			→	—	—	—	<object>には, <grid>, <structure>を記載しなければならない.	複数定義可
				→	id	positiveInteger	オブジェクトの ID [1*] ※以降, 1 ずつ加算	<object>を一意に示すための ID を設定する. 他の<object>の(id)と重複してはならない.	必須
				→	name	String	定義するオブジェクトの名称 [—]	<object>の名前を設定する. 他の<object>の(name)と重複しない方が良い.	
		grid			→	—	—	—	<grid>には, <origin>, <unit>, <dimension>を定義する.
	origin		x	—	—	Double	X 方向の原点までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの X 軸上のオフセット値を指定する.	

fav	object	grid	origin	y	—	Double	Y 方向の原点までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの Y 軸上のオフセット値を指定する.	
				z	—	Double	Z 方向の原点までの距離 [0.0]	ワールド空間の原点から<grid>の原点までの Z 軸上のオフセット値を指定する.	
			unit	x	—	Double	X 方向のセルサイズ [1.0]	<grid>のセル 1 つ分の X 方向の大きさを指定する. 0 以下を指定してはならない.	
				y	—	Double	Y 方向のセルサイズ [1.0]	<grid>のセル 1 つ分の Y 方向の大きさを指定する. 0 以下を指定してはならない.	
				z	—	Double	Z 方向のセルサイズ [1.0]	<grid>のセル 1 つ分の Z 方向の大きさを指定する. 0 以下を指定してはならない.	
			dimension	x	—	positive Integer	X 方向のセル数 [—]	<grid>の X 方向に<voxel>を並べることができる個数を指定する.	必須
				y	—	positive Integer	Y 方向のセル数 [—]	<grid>の Y 方向に<voxel>を並べることができる個数を指定する.	必須
				z	—	positive Integer	Z 方向のセル数 [—]	<grid>の Z 方向に<voxel>を並べることができる個数を指定する.	必須
			structure	→		—	—	—	<structure> には, <voxel_map>, <color_map>, <link_map>を定義する. また, <user_defined_map>を定義できる.
		voxel_map		→	bit_per_voxel	positive Integer	4 / 8 / 16 [—]	ボクセル 1 個が何 bit(16 進文字列何文字)で表現されるかを 4 / 8 / 16 のいずれかで指定する. (それぞれ 16 進文字列 1 文字 / 2 文字 / 4 文字)	必須

fav	object	structure	voxel_map	→	compression	String	none / base64 / zlib /runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib /runlength のいずれかで指定する.		
				layer	—	CDATA	ボクセルの ID を列挙した文字列 [—]	<voxel>の(id)を(bit_per_voxel)で指定したフォーマットで列挙する. <dimension.x>×<dimension.y>分の(id)が 1 行に連結されて記載される. <voxel_map>の(compression)に none 以外を指定した場合, <layer>のデータは指定の圧縮方法で圧縮され, CDATA で格納される.	1 以上	
			color_map	→	—	—	—			
					color_mode	String	GrayScale / GrayScale16 / RGB / RGBA / CMYK [—]	ボクセルに定義される色情報のフォーマットを, GrayScale / GrayScale16 / RGB / RGBA / CMYK のいずれかで指定する.	必須	
				compression	String	none / base64 / zlib /runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib /runlength のいずれかで指定する.			
				layer	—	CDATA	3D モデルの色情報を表す 16 進数文字列 [—]	<voxel_map>に列挙した<voxel>ユニットの色情報を, (color_mode)に指定したフォーマットで列挙する. <color_map>の(compression)に none 以外を指定した場合, <layer>のデータは指定の圧縮方法で圧縮され, CDATA で格納される.	1 以上	
			link_map	→	—	—	—			
					bit_per_link	positive Integer	4 / 8 / 16 [—]	リンク情報 1 個が何 bit (16 進文字列何文字)で表現されるかを 4 / 8 / 16 のいずれかで指定する. (それぞれ 16 進文字列 1 文字 / 2 文字 / 4 文字)	必須	

fav	object	structure	link_map	→	neighbors	positive Integer	6 / 18 / 26 [—]	対象ボクセルの周囲何近傍までリンク情報を保持するかを 6 / 18 / 26 のいずれかで指定する.	必須	
					compression	String	none / base64 / zlib / runlength [none]	<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する.		
			layer	—	CDATA	3D モデルのリンク情報を表す 16 進数文字列 [—]	<voxel_map>に列挙した<voxel>ユニットのリンク情報を, (<neighbors>)に指定したフォーマットで, エラー! 参照元が見つかりません。の順番に列挙する. <link_map>の(<compression>)に none 以外を指定した場合, <layer>のデータは指定の圧縮方法で圧縮され, CDATA で格納される.			
		user_defined_map		—	—	—				
			→	value_type	String	byte / short / ushort / int / uint / float / double [byte]	外部ファイルに格納されているユーザー定義属性の, ボクセル 1 個分の属性値の型を指定する.	必須		
				compression	String	none / base64 / zlib / runlength [none]	外部ファイルに格納されている<layer>データの圧縮方法を none (圧縮無し) / base64 / zlib / runlength のいずれかで指定する. (XML 形式の「.favmapx」ファイル時のみ有効)			
			reference	—	CDATA	ユーザー定義属性を格納する外部ファイルパス [—]	ユーザー定義属性が格納されている外部の.favmap/.favmapx ファイルまでの相対パスを文字列で指定する.	必須		

(例)

```

<?xml version="1.0" encoding="utf-8"?>
<fav version="1.1">
  <metadata>
    <id>bc4affb5-9a53-4de7-9f27-721ef27e8f34</id>
    <title><![CDATA[FAV Ver1.1 Sample File]]></title>
    <author><![CDATA[Fuji Xerox & Keio SFC]]></author>
    <license><![CDATA[CC BY]]></license>
    <note><![CDATA[This is a sample file in FAV format ver1.1.]]></note>
  </metadata>
  <palette>
    <geometry id="1" name="NormalCube">
      <shape>cube</shape>
      <scale>
        <x>1</x>
        <y>1</y>
        <z>1</z>
      </scale>
    </geometry>
    <geometry id="2" name="Plate">
      <shape>cube</shape>
      <scale>
        <x>1</x>
        <y>1</y>
        <z>0.25</z>
      </scale>
    </geometry>
    <geometry id="3" name="Diamond">
      <shape>user_defined</shape>
      <reference><![CDATA[Diamond.stl]]></reference>
      <scale>
        <x>0.98</x>
        <y>0.98</y>
        <z>-1.05</z>
      </scale>
    </geometry>
  </palette>
</fav>

```

```
<material id="1" name="SoftMat1">
  <material_name><![CDATA[Some-soft-materials]]></material_name>
</material>
<material id="2" name="HardMat1">
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]></manufacturer>
    <product_name><![CDATA[ULTRA-HARD/007]]></product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]></url>
  </product_info>
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]></manufacturer>
    <product_name><![CDATA[ULTRA-HARD/006a]]></product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]></url>
  </product_info>
  <standard_name> <![CDATA[JIS K6899-1 ABS]]>
</standard_name>
</material>
</palette>
<voxel id="1" name="soft_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
  </material_info>
</voxel>
<voxel id="2" name="hard_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>0.15</ratio>
  </material_info <ratio>1</ratio>
</material_info>
</voxel / >
  <material_info>
    <id>2</id>
```

```

    <ratio>0.85</ratio>
  </material_info>
  <application_note><![CDATA[HM-H01:Hybrid Hard Material Number 01]]></application_note>
  <application_note><![CDATA[FabAppAttr : application note]]></application_note>
</voxel>
<object id="1" name="SampleObject">
  <metadata>
    <id>cafed8bd-3bd9-4d7a-a67d-2df635d2d8f8</id>
    <title><![CDATA[]]></title>
    <author><![CDATA[Mr. Sample Creator]]></author>
    <license><![CDATA[No rights reserved]]></license>
  </metadata>
  <grid>
    <origin>
      <x>28.5</x>
      <y>-30</y>
      <z>0</z>
    </origin>
    <unit>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </unit>
    <dimension>
      <x>7</x>
      <y>7</y>
      <z>7</z>
    </dimension>
  </grid>
  <structure>
<voxel_map bit_per_voxel="8" compression="none">
    <layer><![CDATA[01010000000000101000000000010100000000001010100000000
000101010000000000010101010100000001010101]]></layer>
    <layer><![CDATA[010100000000000101000000000010100000000001010100000000
000101010000000000010101010100000001010101]]></layer>
    <layer><![CDATA[010100000000000101000000000010101000000000010101000000
0001010101000000000101010101000000010101]]></layer>

```

```
<layer><![CDATA[0101010000000001010100000000001010100000000010101010000
00001010101010000000101010100000000010101]]></layer>
<layer><![CDATA[000101010000000001010100000000010101010000000001010101
00000101010100000000010101000000000000]]></layer>
<layer><![CDATA[00010101010100000001010101010000010101010100000001010101
0000000001010100000000000001000000000000]]></layer>
<layer><![CDATA[00000001010101000000010101010000000101010100000000010101
0000000000000000000000000000000000]]></layer>
</voxel_map>
<color_map compression="none" color_mode="RGB">
<layer><![CDATA[8300258100277600329100176400457c002d5e004a5c004c500059560052
3300753700713000782f007a3100771800900f00991f00891c008c130096
0c009c]]></layer>
<layer><![CDATA[8200269700117400348e001a6300466800405b004e59005055005355005
34000684200663b006d2c007c2400842d007a1700911f00891e008b14009
4100098]]></layer>
<layer><![CDATA[8900209a000e8500237f002a8100277b002d6c003c7700315800515a004e
5300554900604200673d006b390070380070270081230086300079270081
19008f19008f]]></layer>
<layer><![CDATA[9300159800109b000e8100278000288100277f0029720036700039750033
6e003b6f003a58005159004f3e006a4100684100674100673d006b2d007c
2200862000892c007c2100881f0089]]></layer>
<layer><![CDATA[99000f9700119a000e9200169200168a001e7500337300347000386b003d
72003562004759004f6300465b004e6100475d004b55005342006639006f
4300662a007e270081]]></layer>
<layer><![CDATA[9000189000188a001e9700129600128f001988001f8c001c7b002d7a002e
8700218600237700316f00396d003c5c004c6f003a6d003c5800505a004e
5300554f005939006f]]></layer>
</color_map>
<link_map bit_per_link="8" neighbors="6" compression="none">
<layer><![CDATA[00000064c8ff00000000c8ff00000064c8ff00000000c8ff00c80064c8ff
00c80000c8ff00c8006400ff00c80064c8ff00006400c8ff00c8006400ff
00c86464c8ff00006400c8ff00c8006400ff00c86464c8ff00006464c8ff
00006464c8ff00006400c8ff00c8006400ff00c8646400ff00c8646400ff
00c8640000ff]]></layer>
<layer><![CDATA[00000064c8ff00000000c8ff00000064c8ff00000000c8ff00c80064c8ff
00c80000c8ff00c80064000000c80064c8ff00006400c8ff00c8006400ff
```

```

00c86464c8ff00006400c8ff00c8006400ff00c86464c8ff00006464c8ff
00006464c8ff00006400c8ff00c80064000000c8646400ff00c8646400ff
00c8640000ff]]></layer>
  <layer><![CDATA[ff000064c8ffff000000c8ffff000064c8ffff000000c8ffffc800640000
ffc80064c8ff00006400c8ffffc80064c8ffffc86464c8ff00006400c8ff
ffc800640000ffc86464c8ffffc86464c8ff00006400c8ffffc800640000
ffc8646400ffffc86464c8ffff006464c8ffff006400c8ffffc8006400ff
ffc8646400ffffc8640000ff]]></layer>
  <layer><![CDATA[ff000064c800ff000064c8ff00006400c8ffff0000640000ff000064c8ff
00006400c8ffffc80064c8ffffc86464c8ff00006400c8ffffc800640000
ffc86464c8ffffc86464c8ff00006400c8ffffc8006400ffffc86464c8ff
ffc86464c8ff00006464c8ff00006400c8ffffc800640000ffc86464c8ff
ffc86464c8ffffc86400c8ffffc800640000ffc864640000ffc864000000]]></layer>
  <layer><![CDATA[ff000064c8ffff006464c8ff00006400c8ffff000064c800ff006464c8ff
00006400c8ffffc800640000ffc86464c8ffffc86464c8ff00006400c8ff
ffc80064c800ffc86464c8ffffc86464c8ff00006464c8ff00006400c8ff
ffc800640000ffc864640000ffc86464c8ffffc86464c8ffffc86400c8ff
ffc800640000ffc864640000ffc8640000ff]]></layer>
  <layer><![CDATA[ff0000640000ff006464c800ff006464c8ff00006464c8ff00006400c8ff
ff000064c800ff006464c8ff00006464c8ff00006464c8ff00006400c8ff
ffc800640000ffc86464c8ffffc86464c8ff00c86464c8ff00c86400c8ff
ffc800640000ffc86464c8ffffc86464c8ffffc86400c8ffffc800640000
ffc864640000ffc86400c800ffc800000000]]></layer>
  <layer><![CDATA[ff000064c800ff006464c800ff006464c80000006400c800ff000064c800
ff006464c800ff006464c800ff006400c800ffc800640000ffc86464c800
ffc86464c800ffc86400c800ffc800640000ffc864640000ffc864000000]]></layer>
</link_map>
<user_defined_map value_type="float" compression="none">
  <reference><![CDATA[ExternalAttributes.favmap]]></reference>
</user_defined_map>
</structure>
</object>
</fav>

```

図 30 : FAV フォーマット全体の XML サンプル

8. 変更履歴

日付	変更者	変更場所	変更内容
2018.10.11	高橋 智也	表紙	「FAV」に登録商標マークを付与.
		ライセンス	「非営利」を削除. ライセンサーを個人から富士ゼロックス(株)へ変更.
		5. <voxel>	外部の.fav ファイルを参照して任意のボクセル集合を 1 ボクセルとして利用できる機能の説明を追加. 図を更新.
		5.2. <material_info>	0 を指定して材料を配置しない空間を確保する機能を追加. 図を更新.
		5.5. <reference>	外部の.fav ファイルを参照して任意のボクセル集合を 1 ボクセルとして利用できる機能を追加.
		6. <object>	ボクセルの各グリッドセルに対してユーザー独自の属性をマッピングする機能の説明を追加. 図を更新.
		6.1.2. <unit>	1mm が何個のボクセルで構成されるか (vpm = voxel per mm) の計算式を追加.
		6.2. <structure>	ボクセルの各グリッドセルに対してユーザー独自の属性をマッピングする機能の説明を追加.
		6.2.2. <color_map>	階調数の誤表記を修正. 【誤】 512 階調 【正】 65,536 階調
		6.2.3. <link_map>	リンク情報 1 個が何 bit かを表す仕様(bit_per_link)の記載が不足していたのを修正.
		6.2.4. <user_defined_map>	ボクセルの各グリッドセルに対してユーザー独自の属性をマッピングする機能を追加.
		6.2.1. <voxel_map> 6.2.2. <color_map> 6.2.3. <link_map> 6.2.4. <user_defined_map>	各<layer>の(compression)で runlength をサポートする機能を追加.
2019.02.18	藤井 雅彦 高橋 智也	6.2.3 <link_map>	リンク情報を記述する近傍ボクセルの順序を変更. (z,y,x の座標値の小さい順とする) リンクマップのデータ例修正
		4. <palette> 6. <object> 7. まとめ	サンプルプログラム修正