

まとめ

(1) 実習のレビュー

全 8 回の実習で作成したワークスペースは、回を追って単純なものから次第に複雑な構成にするともに、少しずつ新しい知識やテクニックを導入するようにしました。

体験することを通じて FME Desktop の主な機能や基本的な使い方を理解していただくというアプローチでしたが、技術的事項を体系的に説明したものではないので、何を学んだのか、何が重要なのかを整理するのは難しいかも知れません。

それを補うため、実習を大きく 3 つのパートに分けてそれらの要点を掲げますので、参考にしてください。

実習 1 と実習 2 で、ワークスペースを構成する 3 大要素 - リーダー、ライター、トランスフォーマーをワークスペースに追加、接続する方法や、ワークスペースの保存・実行の方法を説明しました。

また、実習 2 では、ライターフィーチャータイプのスキーマ（フィーチャータイプ名、属性名、属性のデータ型）の構成方法も説明しました。

これらはワークスペースを作成、実行するための最も基本的な事項であり、また、データフォーマットの変換だけならば、これらの応用によって行うことができます。

実習 3～実習 5 では、属性名や属性値の変換、属性値に基づくデータフローの条件分岐、属性の結合、属性値に基づく出力先の振り分け方法を取り上げました。

また、実習 4 で取り上げた公開パラメーター（Published Parameter）は、ワークスペースを使い易く、かつ、再利用し易くするための鍵であり、さらに、FME Server で実行したりする場合には、公開パラメーターを適切に定義することが不可欠となります。

実習ではソースデータとして GIS のデータ（Esri Shapefile 形式）を使用しましたが、これらは GIS などの空間データに限らず、あらゆる形式のデータについて、その抽出、統合、移行などを行うワークスペースに応用できます。

実習 6～実習 8 ではさらに、ジオメトリ変換に関する内容を追加しました。FME は実に多種多様なジオメトリの変換機能を備えており、ジオメトリ変換というテーマとしては、実習で行ったことはごくわずかな事例に過ぎませんが、ワークスペースの作成過程では、次に掲げるような汎用的な知識あるいはテクニックを導入しました。これらは、さまざまなシナリオにおいて、必ず役に立ちます。

- フィーチャーの集約（実習 6）
- Group By パラメーター（実習 6）
- ジオメトリ間の空間的關係（実習 7）
- リスト属性（実習 7）
- 座標系の変換（実習 8）
- 属性値の計算（実習 8）

(2) フォーマットとトランスフォーマー

実習で使用したリーダー/ライターとトランスフォーマーを掲げます。

リーダー/ライター

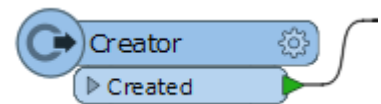
フォーマット名	リーダー	ライター	
Esri Shapefile	*	*	
MapInfo TAB		*	
Microsoft Excel		*	
CSV	*		

トランスフォーマー (アルファベット順)

トランスフォーマー名	概要
Aggregator	フィーチャーの集約
AreaCalculator	面積の計算
AttributeCreator	属性の作成、属性値の設定・変更
AttributeKeeper	属性の削除 (残す属性を指定)
AttributeManager	属性名の変更、属性の削除等
AttributeRemover	属性の削除 (削除する属性を指定)
AttributeRenamer	属性名の変更
AttributeSplitter	文字列の分割
Clipper	ジオメトリのクリッピング
CoordinateRounder	ジオメトリの座標値の丸め
CsmapReprojector	ジオメトリの座標系の変換
Dissolver	隣接するまたは重なり合うジオメトリ (面) の融合
ExpressionEvaluator	数式による数値計算
FeatureMerger	フィーチャーの結合
Inspector	FME Data Inspector への出力
ListConcatenator	リスト属性の要素を連結した文字列作成
SpatialRelator	ジオメトリ間の空間的関係の判定
StringConcatenator	文字列の連結
StringFormatter	文字列の書式設定
SubstringExtractor	部分文字列の抽出
Tester	データフローの条件分岐

ひとつだけ、**Creator** というトランスフォーマーを追加しておきます。

Creator には入力ポートがありません。**Creator** を追加したワークスペースを実行すると、出力ポートからパラメーターで設定したジオメトリを持つフィーチャーが出力されるので、さまざまなトランスフォーマーの動作確認をしたいときなどにリーダーの代わりに使うこともできます。



Creator → [動作確認をしたいトランスフォーマー] → **Inspector**

動作確認をしたいトランスフォーマーが属性を扱うものであれば、テスト用の属性を作成するための **AttributeCreator** を挿入します。

Creator → **AttributeCreator** → [動作確認をしたいトランスフォーマー] → **Inspector**

トランスフォーマーの使用方法はそれぞれのヘルプ (ドキュメント) で説明されていますが、百聞は一見にしかず、文書だけでは分かりにくくても実行してみれば理解できるということは良くあるので、**Creator** を利用して簡易な動作確認をすることをお勧めします。

おわりに

はじめに述べたように、このガイドは **FME** の全ての機能や使用方法についての説明を網羅したものではありません。実習では全く触れなかった技術的事項のうち、使われる機会が比較的多いと考えられるものを掲げます。

Conditional Value Setting (条件別の値設定)
Merging Feature Type (リーダーフィーチャータイプのマージ)
Dynamic Schema (ダイナミックスキーマ)
Dataset Fanout (データセット Fanout)
FeatureReader / FeatureWriter
Custom Transformer (カスタムトランスフォーマー)
Batch Processing (バッチ処理) / WorkspaceRunner
Python Scripting (Python スクリプト)

これらをキーワードにして **FME Knowledge Center** (**Safe Software** 社のユーザーサポートサイト) を検索すると、関連のドキュメントや技術解説記事がヒットします。

FME Knowledge Center: <https://knowledge.safe.com/>

フォーマットについて、実習では数種類のフォーマットしか取り扱いませんでしたが、**FME** は **300** 以上のフォーマットをサポートしています。

ワークスペースへのリーダー/ライターの追加や接続の基本的な方法は共通ですが、ソースデータへの接続方法、フォーマット属性、パラメーターにはフォーマットごとに固有の部分もあります。

未知のフォーマットに遭遇したときは、**Reader and Writer Gallery** によって **FME** がサポートするかどうかを調べ、サポートする場合にはそのフォーマットのヘルプ (ドキュメント) によってリーダー/ライターの仕様を確認することができます。

ジオメトリについて、実習ではベクターしか扱いませんでしたが、**FME** はラスター、ポイントクラウド (点群)、**3D** もサポートしており、それらを変換するためのトランスフォーマーも数多くあります。それらのトランスフォーマーについては、**Transformer Gallery / Categorized** フォルダ内の **3D, Point Clouds, Rasters** などのカテゴリで調べることができます。

上記以外にもこのガイドでカバーしきれなかった事項はたくさんあります。しかし、全てを理解してからでなければ **FME** が使えないというわけではありません。このガイドで学んだことを応用するだけで、十分に実用的なワークスペースを作成できるケースも数多くあります。使いながら少しずつ知識、技術の幅を広げていくというアプローチをお勧めします。

実習では、リーダーの追加からはじめて少しずつ、何段階かに分けてワークスペースを組み立てました。このように段階的に作成したのは、説明し易くするためだけではありません。実務でワークスペースを作成するときでも、ある程度組み立てたところで **Inspector** 等によって変換結果を確認、おかしい点があれば手直しをし、満足いく結果が得られれば次の段階に進む、という段階的な進め方になることがよくわかります。実習では、そのような実務に近い手順を再現することを意図しました。

ワークスペースの作成過程で何度実行してもソースデータを壊すことはなく、変換結果は **Inspector** 等で即座に確認できます。試行錯誤が安全かつ効率的にできるのも **FME** の利点のひとつです。

実務上の課題を解決するためのワークスペース作成にも是非、チャレンジしてください。