

技術紹介

C#を用いた V-nasClair の API 開発

～安定した製品・機能開発に向けて～

PoC of C# API Implementation for V-nasClair Applications.

三浦 武志 *1
MIURA Takeshi窪田 圭志 *2
KUBOTA Keiji青山 貴彦 *3
AOYAMA Takahiko

1. はじめに

KTS は製図ソフトである V-nas を主力製品としています。近年は 3 次元 CAD に踏み込み、V-nasClair を中心とした製品を展開しています。しかし、BIM/CIM 原則適用によって必要となる詳細設計向け製品へ開発リソースを十分に充てられるかが課題となっています。より多くの製品やサービスを提供するには自社制作に固執せず、セカンド・サードパーティーの力を借りることが必要です。そこで V-nasClair の API (Application Program Interface) を整備することを試み、その実現可能性について検証しました。

2. API の概要

本紹介での API は、V-nasClair へ機能追加を行うことができるデスクトップアプリケーションのプログラムインターフェースとします。

他社 CAD の多くは、API, SDK が標準で組み込まれており、ユーザーはデータへのアクセス機能や形状の作成編集機能を利用し、独自機能を追加することが可能です。対して V-nasClair は自社開発で進めており、ユーザーが機能をカスタマイズすることは困難です。このような状況の中、V-nasClair 本体に影響を与えることなく API を作成し、機能追加を行うことができるかを検討しました。

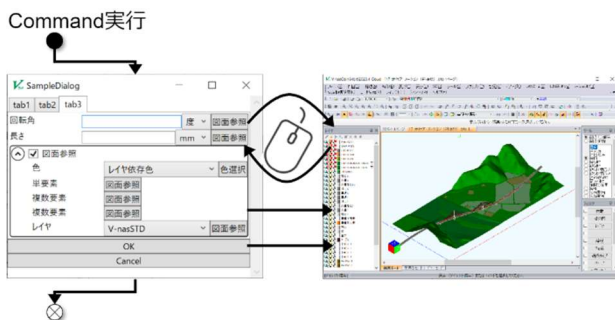


図 1 本 API による処理の流れ

今回の API の検証では、データへのアクセスと形状の作成および UI 操作を分離すること、また UI を統一することを主眼に進め、図 1 の流れとなるようにまとめました。なお、V-nasClair が Windows アプリケーションであること、Microsoft が .NET を推進していることを考え、本 API では C# を開発言語に選定しました。これにより V-nasClair の開発言語である C++ に比べて機能の開発が容易になりました。

3. API による機能作成例

主な利用場面と想定している詳細設計では細かな 3D モデル化を行うことが多いことから、押し出しでの形状作成などの基礎機能が必須になります。これらの形状は、基本的には①アウトラインを作る、②側面を作る、③上下のふたを作る、④閉じる、という流れで作成します。

API ではこのような形状作成操作を定義し、またメッシュ修復機能と体積計算機能、メッシュ切断機能や中心線形の参照についても定義しました。新しい機能を追加する場合、API で定義された機能を組み合わせることで作成します。

例えばスイープ・押し出し・回転体の作成操作と中心線形参照機能を組み合わせることで、図 2 のように 3D 線形モデルを表現することができます。



図 2 3D 線形モデル作成機能例

*1 川田テクノシステム㈱開発本部先端開発部 課長

*2 川田テクノシステム㈱開発本部先端開発部

*3 川田テクノシステム㈱開発本部先端開発部 主任

またメッシュの切断機能を利用し、路面モデルを複数の閉じたポリラインで切断することで、**図3**のように路面標示モデルを一括作成することができます。

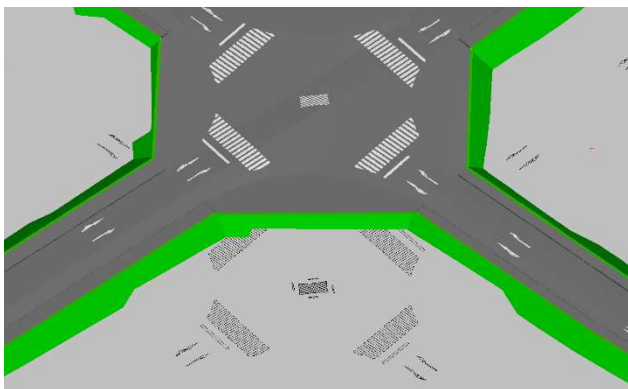


図3 道路標示モデル作成機能例

このように API によって共通化された基礎機能を組み合わせて開発することができ、機能の安定性を保ちながらの生産性向上が期待できます。

4. 既存機能の問題点と API における解消

前項の API を定義するにあたり、V-nasClair には以下のような問題があり、解決する必要がありました。

①文字のように複雑な凹形状（**図4**）では正しくふたの作成ができず、また処理が遅い。②形状の配置方法によっては体積計算が遅く、また処理時間のばらつきが大きい。

これらについて、①三角形検索ツリーの構築とアウトラインエッジ保持方法の見直し、②FEM などにおいて一般的な四面体としての体積計算方法の導入、によって解決を試み、①については**表1**、②については**図5**の形状に対して表のような効果がありました。特に体積計算は、詳細設計では水平でない構造物が増えると考えられ、3D 回転を想定することの重要性が高まると予想されます。



図4 押出形状作成例（赤はエラー部）

表1 サーフェス化性能の検証

	Clairサーフェス化		APIサーフェス化	
	msec/文字	正答率	msec/文字	正答率
ゴシック	93.5	49.3%	6.0	100.0%
明朝	392.4	27.9%	21.9	98.8%

英数記号ひらがなカタカナ(全角半角)と358文字の文で検証した



図5 体積計測対象形状

表2 体積計測性能検証 [sec]

	正対	Z軸20°	3D回転	Y軸90°	計	σ
V-nasClair	1.668	1.185	64.181	2.629	-	27.709
API	-*	-*	-*	-*	0.031	0.009

※5回計測し、APIは4パターンをまとめて計測した

本項に示した対応により、C#であっても詳細設計に必要な性能を満たす API を定義できました。

5. 今後の展望

API として処理を共通化した効果の他、C#を開発言語としたことにより様々なライブラリを利用できるようになります。そのため、これまで社内の一部のみで対応していた機能開発を効率化すること、外部に開発を依頼する際のハードルを下げるすることができます。

また本 API はこれら開発の効率化だけでなく、サードパーティーによるサービスなど、V-nasClair をより広範囲に活用するための足掛かりになると考えています（**図6**）。



図6 V-nasClair 利用機会の拡大

6. おわりに

本機能紹介では主に形状作成機能とその参照にフォーカスして検証を行い、十分に活用可能であることを示すことができました。しかし、その他に属性の追加編集機能を検証しましたが、これについては基本機能としては問題がないものの、ファイルを開く・閉じる、といった V-nasClair 操作と連携をとる必要があることが明確になりました。これらを改善した上で、社内の業務効率化ツールの作成といった活用から展開していく予定です。また、協力会社で開発に利用することや、協業を行う企業とより密度の高い連携アプリケーションを開発することに活用していく予定です。