

## 16. 総括と提言

### はじめに

第2～15の各章においては、1993～2008年度にわたり開発・改良を進めてきた景観シミュレーション・システムの全体構成と各構成要素、および枝分かれバージョンの各機能の統合を実現するために用いられたアルゴリズムに関して解説した。その主たる目的は、本システムを今後更に改良しようとするプログラマ、あるいは本システムのアルゴリズムやライブラリ関数等を、部品として別システムに組み込もうとするプログラマに対して情報公開を行うことにある。

本章では、国立研究機関により開発した公共財としてのオープンソースのシステムの意義と、関連する諸問題について総括すると共に、15年間の経験に基づいて、今後ITに関する同様の、あるいは別の目的のためのシステム開発を行おうとする研究開発機関あるいは建設現場の担当者のための示唆を引き出そうとするものである。

### 16-1. IT化のインパクト

景観シミュレーション・システムの開発に着手した1993年当時は、ユーザーとして想定する現場にPCが導入途上であり、利用状況には事務所間で大きな格差があった。導入されている事務所においては、MacintoshよりはWindows3.1をOSとするものが多く、ビットマップ型のディスプレイ上でマウスを用いて操作するGUIが普及途上にあった。その後急速に進み、ワードプロセッサや表計算ソフトウェアが業務にも使用され、やがてネットワークによる情報交換が行われるようになっていくオフィス環境変化の初期の段階にあった。

#### (1) 真の持続可能なシステム

ITの導入（とりわけ商用）への勧奨は、しばしばデータの持続性をその効用の一つとして掲げる。アナログな実世界をデジタルに計測し、デジタルに記録された情報は、劣化することなしに長期にわたり保存し、コピーし、活用することが技術的には可能となる。例えばCALSは、LS（ライフ・サイクル）にわたる持続的な情報サポートを理念としている。

しかしながら、現実のコンピュータ化・デジタル化の世界は日進月歩・ドッグイヤーの世界である。今日作成したCADデータは、5年後には遠い過去のデジタル古文書と化す。

例えば、本システムの開発に着手した1993年頃の官庁営繕分野における技術研究会に参加したことがあるが、設計時に作成されたCADデータを長期にわたり保存・活用するための媒体として、8インチフロッピーディスクを用いるべきか、それとも3.5インチ光磁気ディスク(MO)を用いるべきかが真剣に議論されていた。現在、これらの媒体に記録されている情報を読み出して利用できる環境(ドライブ等)は既に希少である。

これは根底からの矛盾であり、本来は建物に関する情報を保管する使命を帯びた「国立

公文書館」のような組織機関こそ検討されるべきである。真剣に情報を長期にわたり保管することを考えている者は、実績のない全く新しく開発されたシステムへの情報の載せ替えには、たとえそれが「情報の永久的な維持管理」を標榜するものであっても躊躇するに違いない。

理論的・技術的に可能であり容易であるはずの情報の長期的な維持管理を実務的に実現するためには、少なくとも保管のための物理的な媒体が十分な耐久性を有し、データ形式などが（たとえ時代遅れではあっても）永久不変であることが求められる。

同じことはソフトウェアに関してもあてはまる。ある技術を具体的な処理システムとして実現したソフトウェア資産は、劣化することなく継承発展させることが可能であるはずである。しかしながら、ハードウェアや OS の急速な進化により、同じプログラムを実行できる動作環境が失われる。継続的に使用するためには、その都度の環境変化に適応するために継続的にソフトウェアのアップデートへの投資を続けなければならない。

100年後の歴史家は、17世紀の和紙に筆で書かれた若干の虫食いのある地方文書を現在と同様に解読できるであろうが、倉庫に安全に保管されている20世紀末のハードディスク装置の中に記録されている、当時のワードプロセッサで作成された文書を読み出して解読するためには、かなり苦勞することが想像される。

つまり、本来永久不変である筈のデジタル化された情報が、現時点では、絶えずアップデートと媒体変換し続けなければ維持継承できなくなっている、という意味で、あたかも稗田阿礼以前の口述伝承のような世界に戻ってしまったかのごとくである。

しかしながら、IT に関する研究開発においては、最先端のデバイスや技術や利用技術への関心に隠れて、上記のような問題に関する議論はあまりに少ないように感じられる。

## (2) 途上国におけるデジタル化

筆者は、1984年から継続的にインドネシア共和国の公共事業省人間居住研究所(英語名 Research Institute for Human Settlements, Ministry of Public Works、現地名 Pusat Penelitian dan Pengembangan Permukiman, Departemen Pekerjaan Umum)に対して技術協力のために継続的に派遣される機会があり、都市住宅政策分野に関連するその都度のテーマを具体的な課題としつつ、25年間にわたり定点観測的に同研究所における最初のコンピュータ導入から、最近のネットワークや GIS 導入に至るまで助言と観察を続けてきた。その間、記憶媒体は、紙から5インチフロッピーディスク、3.5インチフロッピーディスク、CD-R、スティック・メモリと変化してきた。

初期の状況は、日本建築学会などのような論文をデータベースとして管理する機関が存在しない同国において、国立研究所が発行する機関誌「Masalah Bangunan」（「建築問題」と訳せる）が唯一の専門誌であり、テンポラリーに作成されるセミナーの「Proceedings」が情報源であった。研究所が備品として維持管理し専門職員が操作する「印刷機」は、同研究所の重要なリソースであり、社会的使命を果たしていた。これに概要が報告される研究成果自体は、個別の報告書（多くは国際援助として実施されたプロジェクトの成果であ

る)として記録されるものの印刷部数は少なく、報告書の質は玉石混交であって、整理保管の体制が不十分な図書室等においては、価値のある文献である程よく貸し出され、返却されないことにより失われていく。担当官吏が個人的に保管している報告書等も同様で、「本を貸す馬鹿返す馬鹿」のような状況であった。見知らぬ来訪者に「その問題に関しては調査報告書がある」と言えば、「見せてくれ」と言われる。見せると「貸してくれ」と言われる。貸すと返って来ない。このような状況であったから、報告書を保管している関係官吏は、その所在についてなかなか語らない。結果的に、価値のある文献ほど、アクセスが困難であり、その事が知的財産の社会的普及・活用を妨げていた。

このような状況を変えたのが、1990年代に入ってからのコピー・サービスの急速な社会的普及で、低価格の小型コピー機を用いた零細なコピー店がまず街中に急増した。人件費が安いことから、コピーし簡易製本まで安価に依頼することができるようになり、これにより原本を長期に借り出すことなく情報を入手することが技術的に可能となった。これは情報公開・流通を大きく促進した。

1987年頃から、IBM-PC/XTをモデルとした台湾製のコンパチブル機(ハードディスクなし)が約10万円程度で市販された)が徐々に普及し、WORD-STARにより入力された文書ファイルを、5インチ・フロッピーディスクにコピーして入手することが可能となった。日本から供与したNEC-9801Eとは媒体による互換性がなく、RS-232Cクロスケーブルを製作して接続し、ソフトウェアを用いてファイルを転送した。

次の転機は1997年のアジア通貨危機に訪れた。ドルや円に対する為替レートが約5分の1に急落し、輸入に依存していた文具や紙製品の価格が急騰した。これにより、紙のコピーやFAXを使用することが急減し、必要な情報をフロッピーディスク等により交換し画面で表示して用を足すことが一般的となった。つまり、アジア通貨危機は研究所のペーパーレス・オフィス化を一挙に進めた。日本では現在でも大量の紙コピーの資料を会議で配布するスタイルが継続しているが、これは紙が安いからできる贅沢であると言える。

2000年代に入ると、安価なCD-Rが普及し、これを用いて大規模なCADやGISソフトウェアの海賊版が電気街で販売されるようになった。概ね1000円程度で入手可能なこれらの製品は、学生などにも広く普及した。このような状況下で、地図や建築設計図のデータが、紙ではなくCD-Rを媒体として交換されるようになった。これらのソフトウェアの価格は日本では25~100万円程度であることから、日本の学生は現在でも大学の計算機センターで触れることはあっても、在宅やテイクアウトで日常的に触れることはまだ実現されていない。途上国では建築設計事務所などにおいても、CADソフトやGISソフトを用いて作業が行われることが直ちに一般的となり、製図版やドラフターは殆ど見られない。

但し、途上国におけるこのような情報流通面におけるデジタル化とは対照的に、情報の長期的な保存管理という点に関しては、未だに非常に脆弱である。コンピュータ・ウィルスが蔓延していることから、ハードディスクに集約・蓄積されている情報の安全性は低い。「10年前のセミナー論文のデジタルデータ」を現地で探すことはかなり困難である。つま

り、情報のデジタル化・ペーパーレス化が進んだものの、長期的な保存・アーカイブ化に関しては（もし社会的な必要があるとすれば）大きな課題を潜在的に抱えているのが現状である。しかし、少なくとも「簡単にコピーできる」ことが情報の社会的普及に加え原資料の中期的保存に貢献していることは確かである。

### （3）三次元データの意味

設計図から出来形を想像することが職業人の役割であるとするならば、三次元 CG 技術は、変化する技術環境下で、その想像力を補助する意味を有している。技術と様式が安定していれば、簡単な板図のみで住宅の施工が可能であった時代は遠い過去ではない。

建築設計の教育においては、パースの作成がカリキュラムに組み込まれており、平面図・立面図・断面図から対応する代表点を拾い、作図により透視図上に点の位置を求め、それらを基点としながら、絵画としてパースを描画する訓練が行われた。筆者が学生であった1971年頃には、教育用電算機が使用可能となったため、射影変換のプログラムを組み、平面図・立面図から抽出した代表点の XYZ 座標をパンチカードで入力し、LP 用紙に打ち出された変換結果（UV 座標）を図にプロットする方法で作業能率と精度の向上を試みた。

一方、土木分野では、専ら力学的な基準に基づいて形状が決定され、それが風景に与えるインパクトについては、担当者も事前に十分に把握していないような状況が、1993年頃まで続いていた。しかし、シビック・デザイン運動の高まりから景観への影響に関する関心が高まり、コンサルタントが作成する分厚い設計図書にパースが添付され始めていた。

紙の上に描かれた設計図は、二次元の図面による表現ではあるが、設計対象物の三次元形状を記述している。パースを作成する作業は、設計とは別の職能として補足的に行われていた。

三次元データによる画像処理は、この部分を大幅に迅速化し、三次元データから、任意視点のパースを瞬時に作成することを可能とした。視点を少しずつ移動させながら、1秒間に数十回パースを再描画することにより動画が生成する。更に、同じシステム上でモデリング、すなわち三次元データの編集を行うことができれば、表示結果を見ながら、設計内容自体を変更する、SBD（Simulation Based Design、シミュレーションに基づく設計）が景観に関して可能となる。

景観シミュレータに、価値判断の機能を持たせるべきか否かについては、未だに議論がある。景観シミュレーション・システムの開発にあたっては、まず三次元データのモデリングとパース表示のためのコストと時間を劇的に小さくすることにより、状況を変えようとした。これは、かつては専門職人に依頼しなければできなかった写真撮影が、測距や照度計測機能を備えたカメラの出現により、家庭の主婦でも可能となった事や、ワードプロセッサにより、印刷物の最終レイアウトまでが原稿執筆者に可能となった事に比較される技術的環境の変化をもたらすと期待したものである。即ち、当面の選択肢が景観に対してどのような影響を及ぼすのか、簡便にチェックできるような作業環境を設計担当者に対して提供することにより、カメラを手にした家庭の主婦やワープロを手にした作家のように、

最終的な出来形を認識しながら作業を進めることができ、同時に感性を磨くことにもつながるであろうと構想した。

景観検討の対象となる領域に関して三次元データを整備するためには、単に設計対象物のみならず、地形や周辺市街地などに関するデータを効率的に取得・作成することが必要であり、数値地図の整備や様々の立体的計測技術の発達、更には設計図書のデジタル化など、そのプロセスは現在進行中である。とりわけ、平成18年度から開始された電子納品の三次元化の取り組みには、大いに期待されるものがある。

景観設計を入口として、設計対象物及び周辺環境の三次元データが次第に整備されるならば、そのデジタルデータは温熱環境、音響環境、電波環境などの領域にも利用可能である。また、道案内などの様々の行政サービスへの応用の展開も期待されている。

#### (4) サーバー側のアプリケーションとの連携

2001年に開発した「まちづくり・コミュニケーション・システム」においては、景観シミュレータ `sim.exe` をローカルにセットアップした作業環境においてネットワーク上のサーバーからダウンロードした `LSS-S` ファイルを起点として、モデルを記述する `LSS-G` ファイルやテクスチャ、マテリアル、増補された外部関数実行形式など、関連ファイルをダウンロードし、景観として表示することができる。

サーバーからデータを配信する場合、配信されるファイルは必ずしも固定的である必要はなく、リクエストに応じて、サーバー側が保持するデータベースから選択されるファイルや、サーバー側での計算処理により動的に生成するファイルであってもよい。また、クライアント側での編集操作によりローカルに作成された三次元データをサーバーにアップロードすることも可能とした。

これらのサーバー側の機能により、多様なサーバー側の機能を発展させることが可能であり、そのための出発点となる基本構成を上記のシステムが提供している。

セキュリティを確保したサーバー上に基本データ、アプリケーション、セットアップ、ソースコード、関連ドキュメントを置いて配信するという体制は、デジタルデータの継続的な維持保管の問題を最終的に解決するものではないが、少なくともデータの保存継承の問題を、サーバーの維持管理の問題として集約整理することにはなると考えている。

## 16-2. 公共財としてのソフトウェア資産

### (1) 知的所有権

知的所有権は、それ自身が著作物として登録可能であり、昭和62年から、財団法人ソフトウェア情報センター(SOFTIC)において、マイクロ・フィッシュの形での登録・記録保存が行われている。[ソフトウェア](#)による情報処理が、[ハードウェア](#)資源を用いて具体的に実現されていることを要件としてソフトウェア特許としての登録も可能となっている。景観シミュレーション・システムの場合には、公共財であって、これを使用しようとする者に対価を請求する意図が無いことから、そのような知的所有権保護のための手続きを一切行

ってこなかった。しかしながら、他者が本件に関する知的所有権を主張することや、公開し自由使用に供するソフトウェアを、メディア代以上の価格で再販売されることは好ましくないと考えた。そのため、1996年以降は、本システムの開発成果を公知の事実とすることで、他者が知的所有物として登録することを防ぐために、その都度技術資料として公開した(付録C参照)。併せて、1996年に建設省建築研究所のWEBサーバーから、また2001年以降は国土技術政策総合研究所のWEBサーバーからダウンロード・サービスを開始した。但しダウンロード・サービスを開始した時点では、通信速度に限界があったため、一時回線が混雑する状況も生じた。

## (2) 公的機関による開発の意義

大学などで研究開発が行われる場合には、公的資金により開発されたコア技術をもとに、投資を募り、更に応用開発を進めて製品化した上で、ベンチャービジネスとして、販売し収益を上げるということが行われる。

公的機関に限らず、ある組織が内部使用のためのシステム開発を行い、成果を公開せずに自社システムとして各地方支局などに導入することはありうる。必ずしも公開する必要はない。

自社システムを構築する場合には、必ずしもゼロからの開発を行う必要はなく、市販ソフトをカスタマイズするような方法が、より早い。景観シミュレーション・システムの場合には、開発に着手した時点で、現場に導入可能な価格で提供された製品などが存在していなかったために、ゼロからの開発を余儀なくされた。

開発のためのコストおよび工期の縮減に資する補助的なシステムは存在していた。例えば、ユーザー・インターフェースをグラフィックな開発環境でデザインした上で、各種プラットフォームのためのソースコードを出力するための開発システムも検討したことがある。しかしながら、最終成果であるアプリケーションを各現場で導入するに際して、ランタイムのライセンスを購入しなければならない、というビジネス・モデルであったために導入を断念した。

国の機関が民間に外注してソフトウェアを開発し、内部的に利用するような場合、当時その知的所有権に関しては必ずしも明確ではなかった。納品された実行形式に関して、仕様通りの動作が実現されていれば十分とされ、ソースコードの納品を求めないような場合も存在していた。本システムの開発にあたっては、建築研究所及び土木研究所からの外注により多くの部分が開発されたが、公開を前提とするソースコードの納品をもって成果とする契約方法とした。開発を受注した業者は、開発された技術を他の目的のために流用することが行為として可能であるが、これを契約上の守秘義務として縛ることは意義も実効性もないと考えられた。むしろソースコードをオープンソースとして公開することにより、他の者と全く同等の条件で開発成果を別の目的のために活用することが可能となる。そのことは、開発の上位目的である、本システムを具体的に活用した景観検討業務にとって何らの妨げになるものではない。

### (3) 公開の目的

オープンソースとして公開することの消極的な意味は、開発を担当した研究所が、持続的にメンテナンス（デバッグや、OS の変化への対応など）を行う当事者能力を有さないため、デバッグ等をユーザー側にゆだねる、という点にある。

一方、公的機関が国費で開発を行うことは、民業圧迫にあたるという批判も存在していた。ソースコードが公開されなければ、無償ソフトであってもその技術を民間の技術開発に用いることはできないであろう。ソースコードの公開が行われていれば、民間の技術開発は、公開技術を出発点として進めることができる。ソフトウェア開発が、商業的利益を目的とするものではなく、最終的には景観の向上を目的とするものである以上、民間の関連技術開発は推奨される必要がある。

1995 年当時、ソフトウェア分野における技術開発とそのための投資が、市場経済的に合理的な水準まで行われなかった理由として、違法コピーなどにより開発への投資が回収できないことが挙げられていた。このことを打開するために、製品に違法コピーを防ぐためのプロテクトをかける方法や、頻繁なバージョン・アップを繰り返す方法などが採用されていた。これはユーザー側から見ると、実用的な利用にとって必ずしも快適な環境ではない。

しかし、国際協力の現場で目の当たりにすることは、発展途上国においてプロテクトをはずした CAD や GIS 製品が低価格で販売され、アンダーグラウンドのマーケットが形成されている、という事実であった。経済水準の低い途上国では、メーカーが販売収益を上げることが見込まれないために、知的財産権を守るためのコストをかけることを断念し、違法コピーの横行を黙認しているように見受けられた。このため、例えば学生における CAD や GIS ソフトの普及率は途上国においてきわめて高いという逆説的な現象が生じている。新興国が経済発展を遂げて、正規ユーザーとしてソフトウェアを購入できるようになった次の世代においては、IT が急速に社会的に普及する下地は十分に形成されている。そのための準備としては、違法コピーの放置はむしろ布石ともなっているように見える。

### (4) 民間商用ソフトウェアとの関係

開発に着手した 1993 年当初、いくつかの CAD メーカーなどから営業を目的とする訪問があった。メーカー側の意図は、研究所におけるシステム導入であった。当時、建設省中国技術事務所においては、一つのシステムを導入し、これを運用して管内の現場における景観検討を行う試みも行われていた。しかしながら、本システムを開発するに当たって想定した業務形態は、個々の開発現場において、担当者あるいは業務委託を受けたコンサルタントが日常的に景観検討を行うことであった。民間で開発されている CAD ソフトウェアのライセンスを買い上げた上でオープンソース化し、これに基づいて機能開発を進める方法も検討したことがある。旧バージョンであれば応じて良いという社もあったが、ライセンスの価格において折り合わなかった。

知的財産のプロテクトを前提としつつ継続的収益を目指す民間ビジネスと、公開を原則とする公共財の領域の衝突や矛盾を回避し、連携を図るためには、官民共同研究によるデ

ータ共有の仕組みが有効である。初期の開発が概ね終了し、バージョン 2.01 を公開開始した 1996 年時点で、官民共同研究を開始した。公募の結果、空撮写真から現況市街地データを作成する技術と、地上写真からテクスチャ付建物三次元データを作成する技術が対象となった（詳細は文献 16 参照）。いずれにおいても、考え方として、民側の商用ソフトウェア（有償）や測量サービスないし測量結果としてのデータ（有償）と、官側のオープンソースのフリーウェアの中間に、共同研究成果としてのコンバータ（ソースコードを公開しないフリーウェア）を置く、という方法を採用している。

民間のソフトウェア開発は、知的財産のプロテクトを確保しつつ最先端の技術を追求する方向を志向する。そのことが弊害としてデジタル情報の恒久的保存活用の困難をもたらしているとするれば、公的な IT 化はユーザーとしての立場のみならず、研究開発において競争を避け、調和・協調を図りながらも、上記のような欠陥を補完するような役割を果たすことが必要である。既に商品としての寿命を終えたような製品で、基本的なアルゴリズムを有するものについては、今後検討されても良い。

#### **（5）外注の効率性：プログラマとの緊張関係**

公共財としてのソフトウェアを開発する方法（業務形態）として、いくつかの異なる体制が考えられる。事業官庁においてインハウスで開発を行う場合には、成果としてのソフトウェアの販売収益が目的ではなく、組織内部の時間節約・効率向上と、それを通してもたらされる社会的便益が上位目的である。

実際には、ソフトウェア開発の多くの部分は、民間企業に対する外注を通して行われる。発注者側においては、公共工事と同様に、一定の工期・予算の中で要求性能を実現するプログラムを完成させることを期待しており、不完全な部分は受注者の責任において修正することが当然と考えられる。しかしながら、これまでに存在していなかったような新しい技術を投入するソフトウェア開発は、研究開発の委託のような性格を帯びる。最終成果物がどのような機能を果たし、どのように使えるのか、操作体験のない発注者側において、契約時点では必ずしも明確に認識されているわけではない。このため、納期に近い打ち合わせ段階でデモンストレーションが行われた段階で、初めて発注者側に明確な使い方に関するイメージが形成され、それが追加要望のような形でリクエストされるような状況が生じうる。一方、受注者の側では、当面期待されているソフトウェアの動作（それは打ち合わせの結果修正を要求される可能性がある）を実現することが短期的目的となるため、基本的なデータの整合性の確保（とりわけ使用済みのメモリブロックの解放処理等）といった、外見では見えない部分の完成は後回しになる傾向が生じる。納期に近づいてから、要求仕様に関する認識のずれが明確化し、修正が行われる場合には、納品されるソフトウェアは、「とりあえず動いている」程度の完成度レベルのものとなる。

このようなソフトウェア資産に関して、安定した確実な動作を行うように改善していくためには、相応の手間とコストがかかり、受注者の品質保証責任の範囲内だけで実現することには限界がある。しかしながら、既に納品され、外見上の動作は実現している作品の



内的な改善・修正（外見上は大きな変化を伴わない）は、別業務としては成立しにくい。このため、継続的な業務発注において、改善（機能の追加等）を一義的な目的とする新たな業務が、通常同一の業者に対して行われ、その中で前回業務の未完成部分に対する修正も行われることが期待される。しかしながら、同一ソースコードに対して手を加え、完成度を高める作業と機能を追加する作業（完成度を低める方向）を両立させることは容易ではない。動作が不安定な既存部分に見切りを付け、ほぼ同一の目的のための別のデータ形式や処理ルーチンが追加されるならば、更に状況は複雑化する。

景観シミュレーション・システムの場合には、結果的にこの部分の作業のほとんどはインハウスで行われてきた。Ver.2.09に向けた統合化においては、完成度を追求する基幹部分と、機能追加のためのプラグイン DLL を分離する、というアプローチで整理した。

ある段階から機能追加を民間へバトンタッチしていく方法がありうる。景観シミュレーション・システムの場合には、現在がその時期ではないかと考えている。

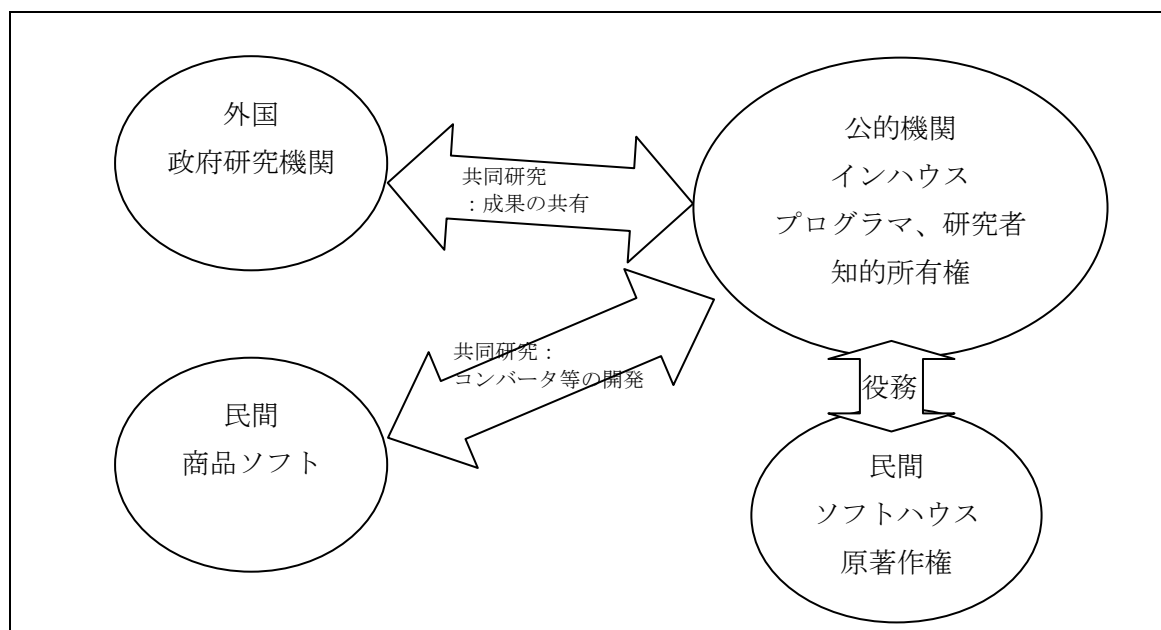


図16-1：景観シミュレーション・システム開発における連携

## (6) 国際協力

国際協力の入り口として、ソフトウェアの翻訳を行う必要がある。ライセンスが制約された商用ソフトウェアをベースとして公的機関の間で国際協力を行うことは困難を伴う。一方、オープンソースであることにより、制約条件が無い。公開WEBサイトからダウンロードすることにより、任意で参照することは可能である。しかしながら、国民が納めた税金を財源として開発された知的財産を外国に対して無償で提供することは不公平と考えられたため、日韓科学技術協力協定に基づく共同研究を実施した際には、研究所間で **Record of Discussions** を取り交わし、相手国側が同一技術の開発を重複して繰り返すことを避ける代償として、提供した技術の上で開発された成果を共有（還元）することを謳った。ソースコードをベースとして実際に実施した初期の翻訳移植作業においては、開発環境の違

いなどにより発生した障害（開発環境のバグや、ソースコード中の潜在バグの顕在化など）に対処する必要があり、言語専門家による単純なリソースやヘルプファイルの翻訳移植だけでは済まず、両国のプログラマによる協力が必要であることが経験的に分かった。今回リリースする多言語版においては、このような問題は解消している。

### 16-3. ユーザーとの共進化

#### (1) 現場の担当者

ワードプロセッサのように、毎日の業務に使用するものではない。工業高校程度の教育を受けた職員が、月に1度程度の頻度で、説明会の前1週間の間操作する、という使用が行われた。このような使用環境においては、基本的なメニュー構成を変更せず、追加機能は奥の方に作りこんでいく戦略が適切であると考えられた。民間の商品ソフトウェアは、プロフェッショナルなユーザーが日常的に使用することを想定して、ショートカット等の能率向上のための機能がふんだんに盛り込まれているが、これらは偶に使用する程度のユーザーにとっては、わかりにくく忘れ易い。また、日常的に使用するワードプロセッサ等においても、バージョン・アップに際して、追加された新しい機能が前面に押し出され、ルック・アンド・フィールが大幅に改定される場合があり、慣れるために時間を要する場合がある。

#### (2) 学生

筑波大学では、MiniCADを教材として学生の教育を行っていた。彼らに景観シミュレータを用いて作業を依頼すると、操作性に関して、教育を受けたシステムとの違いにおいて意見を述べる場合が多かった。最初に習得したシステムが重要であることがわかる。業務用のソフトウェアに習熟している学生等の場合には、景観シミュレータのメニューから明示的に機能を選択する方法は、プロ仕様のソフトウェアにおけるショートカットやコントロールキーによる迅速な操作と比較して手間がかかるという印象も与えるようである。

#### (3) 子供

つくば科学フェスティバル、土木の日などの公開の機会を積極的に活用して、操作を体験して頂いた。このことを実現するためには、基本的な操作（インターフェース）が単純である必要がある。最新の例では、敷地を自由形状として作成した上に、データベースから選択した住宅を配置し、そこに様々な遊具や樹木などを点景として配置するような操作が好まれる。このような操作に関しては、指導員（職員や高校生など）に対して事前に30分程度のレクチャーを行い、直ちに来訪者との対応に入っていただくことが可能であり、助言しながら子供に選択肢を示して操作を進めてもらったり、習熟に応じて自分で操作してもらったりすることができる。基本的な視点移動や、配置等の機能選択方法が単純であることが重要な条件である。

#### (4) 操作性に関する慣習

景観シミュレータの操作性は、「初めてこの種のシステムに触れる」ユーザーを想定して

開発された。メイン画面の上に、ファイル、編集、表示、ツール等の基本的なメニューを示す点、視点移動は機能を明記した操作ボタンによる点、モデリング操作は、まず画面でオブジェクトを選択してから、それに対する操作を選ぶ点などである。この点は、プロフェッショナル向けの本格的な CAD や GIS が最初に一定のトレーニングを必要とする状況とは異なっている。展示会や体験教室においても、高度な処理を含まない基本的な編集操作に限定したコンテンツと体験内容に限定するならば、高校生アルバイトなどに約30分間の事前トレーニングを施すことにより、指導員として直ちに来場する小中学生に操作指導することが可能である。

一方、最近の土木・建築系学生は、それぞれの大学で特定の製品による教育を受けている。彼らの感想（改善提案）は、概ね既に習熟しているソフトウェアとの対比として述べられることが多い。このことは、多言語版を海外で使用する場合でも同様である。

#### **(5) 操作における陳腐性の価値**

ソフトウェア商品が新たなユーザーを獲得するために、その独自性・新奇性を強調する場合がしばしば見られる。このことはバージョン・アップに際しても生じうる。一方、既存製品のユーザーにとっては、操作手順の変更は戸惑いの原因となる。とりわけ道具としてソフトウェアを、毎日ではないが一定の頻度で使用しているユーザーにとっては、度々再学習が要求されることは負担である。景観検討や現場説明の業務であれば、数ヶ月に一度、数日から1週間程度の作業時間を投入する人が多いようである。

一例として、位置の指定や形状生成における画面クリックと座標数値入力を考えてみる。形状生成において、概略の図形を生成するだけであれば、画面上のポイントをマウス・クリックすることにより座標指定する方法が能率的で、素早くデータを構築することができる。ソフトウェアの性能のデモンストレーションにおいては、効果がある。一方、既に設計図が存在するような場合には、点の位置を座標値により指定する方法が速い。そこで、多くのダイアログでは、このいずれの方法でも点の位置が指定できるようにした。画面クリックで点の位置を指定した場合には、座標値を表示するエディット・ボックスに座標値が表示される。一方、座標値を入力し、マウス・カーソルを他の場所でクリックすることにより、画面表示もその座標値に対応した位置に移動する。

マウス操作で指定されたパラメータを直ちに形状表示に反映させる方法は、全体のデータ量が小さい場合には有効であるが、データが大きくなってくると、反応の遅さとして認識されるようになることにも留意する必要がある。

このような別の一例として、カラー編集画面がある。初期のバージョンにおいては、マテリアル編集ダイアログにおいてスライドバーを動かしてユーザーが変更した新たな色彩を、同ダイアログ内のサンプル画像に反映させるだけでなく、メイン画面で選択された編集対象のオブジェクトにも同時に反映させる動作としていた。しかし、一つの市街地全体などが編集対象となっているような場合、メイン画面の再描画に時間を要するために、スライドバーがマウス操作に追従して軽快に動かなくなり、非常に重い動作となる。この

問題を解決するために、ユーザーがスライドバーを動かした後、一定時間スライドバーが動かなかった場合に初めてメイン画面にその結果を反映させるという処理を追加した結果、軽快な操作を確保することができるようになった。

公共財として開発するフリーウェアにおいては、バージョン改訂に際して、操作性を大きく変更することによって、改良を強調表現する必要は一切なかった。このことは継続的なユーザーのニーズと一致していると考えている。

#### (6) バグ発見の手がかりとしての体験教室等の効用

景観シミュレータを繰り返し操作する者は、開発担当者も含め、無意識にリスクのある動作を避ける傾向がある。また、開発において想定された操作手順に忠実に従う傾向がある。これに対して、初めてシステムを操作する展示会来訪者や、初めて操作する非常勤職員などは、開発者が予想していなかった操作を行うことがしばしばある。その際の戸惑いや、理解の障害を認識することは、マニュアルやヘルプを改善する手がかりとなる。また、バグによるシステムのフリーズやプログラムの異常終了の経験は、そのようなリスクを伴う処理を無意識に避けるように学習させることとなり、バグの温存につながり易い。そのような意味で、初めてシステムに触れるような人々を対象とした体験教室等への参加は意味があると考えている。また、そこで生じた事柄を記録し、以後の改善に役立てていくことも有効である。

### 16-4. データ形式

本システムの開発に着手した時点では、既存のデータ形式としては DXF 形式が存在しており、二次元 CAD における流通（例えば、製図に使用する衛生陶器製品の形状データのメーカーから設計事務所への提供）に使用されていた。しかしながら、景観検討に必要なテクスチャやマテリアルを付したオブジェクトの記述には不十分であった。CAD ソフトウェアは当時、プロッターを駆動するために、製図における線を記述することが基本となっていた。複数の面から構成されるオブジェクト単位での操作を表現するデータ形式とはなっていなかった。景観シミュレータにおいても、「線」の記述は可能であるが、実質的にこの部分を用いることはまれである。一方、CG では、Silicon Graphics 社が中心となって、テクスチャを張り込んだオブジェクトの出力表示処理を標準化する OpenGL を提示し、高価なグラフィック・ワークステーションのみならずローコストな PC でグラフィックス・ソフトを開発する環境が提示され、CAD のデータ形式を拡張して三次元 CG と一体化することが課題となっていたが、そのようなデータ形式がまだ存在していなかった。

CAD データの流通は長年の課題であったが、民間のベンダーを集めたフォーラムにより共通の交換形式を定め合意に至ることは、当時難航していた。公共的に使用されるデータ形式を定める場合、特定の民間商用ソフトウェアで用いられている形式をそのまま採用することはできない。必要最小限の独自のデータ形式を定め、公開した上で、実用に向けては、普及しているソフトウェアとの互換を図るコンバータを作成することで対応すること

が適当と考えた。

当時、国土地理院において、「建設技術評価制度」を運用して、ステレオ空中写真解析技術の評価検定を行っていた。その際に、各社に対して、指定したフォーマットでのデータの提出を求めている。このため、技術を有する各ベンダーは、このデータ形式で出力する機能（コンバータ）を作成し、評価を受けた。従って、景観シミュレーションのためのコンバータを作成するためには、この共通の評価形式をハブとして、これに対応することで大幅に手間を省くことができる。このことが参考となった。このような目的のためのデータ形式は、パフォーマンスや拡張性よりもむしろ、単純明快性が求められる。例えば、公共事業における設計データの納品のために目的を限定した単純明快なデータ形式が定められれば、三次元オブジェクトに関するコンバータ作成の労力は大幅に軽減されると考えられる。

民間各社は、デファクト・スタンダードとすることが利益拡大につながる、という動機を有する反面、一定のデータ形式を固定的に定めることは、日進月歩の技術革新の中で、新たな状況に移行することの障害となる恐れがある。実際に、GISのSHAPE形式やCADのDXF形式は、論理的に一貫しない形式の拡張や、ヘッダー部分で識別した上で、異なる構造のデータを並存させるような仕様となっている。

景観シミュレータの開発の目的はこのようなデータ形式を定めるということには無い。そこで、将来のそのようなデータ形式の成立に期待しつつ、当面する景観検討ニーズに対応できる必要かつ十分なデータ形式を定め、対応することとした。

その基本的な考え方は以下の通りであった。

- ・地物を構成する単位となるオブジェクトを、表面の集合により表現する（グループ）。
  - ・オブジェクトを階層構造により表現し、上位のオブジェクトにリンクする際に、相対的な変位（平行移動、回転、スケール）を定義できるようにする。
  - ・リンクを複数設定することにより、群として線的・面的な配置を表現する（例：街路樹や山林等）。これによりファイル・サイズを節約する。
  - ・XYZ座標から成る三次元の位置情報を有する座標(COORD)を定義する。
  - ・頂点には座標の他に、テクスチャ座標、法線ベクトル、カラーを記述できるようにする
- (VERTEX)
- ・面(FACE)及び線(LINE)を、頂点列として定義し、その順番で面の表裏を決める
  - ・面の表面光学特性として、カラーとテクスチャを直接記述し編集することとする。
  - ・グループや面に対して、別途定義されたマテリアルを設定できるようにする。
  - ・マテリアルの定義に、経年変化が記述できるようにする他、カラーやテクスチャの他に、鏡面反射率や輝度等も定義できるようにする。将来材料のCG表現技術が高度化した場合であっても、マテリアルを用いた地物データであれば、マテリアル定義を高度化することにより、地物データを変更することなく対応可能である。

## 16-5. システムの性能

### (1) 処理時間

Sim.exe の処理時間には、CPU 性能に依存する部分と、グラフィックス性能に依存する部分がある。具体的には、ファイルをロードするのに要する時間は前者に依存し、視点移動は後者に依存する。外部関数（パラメトリックな部品）の処理には一時的なファイルを生成するため、これに加えて、ディスク・アクセス時間が影響する。ファイルのロードに際しては、頂点や面やグループの名称などをテーブルで管理するため、保存よりもはるかに処理時間がかかる。頂点や面を定義するために使用する名称は、GROUP\_FACE コマンドにより確定するため、それ以後に同じ名称を用いて再定義しても過去に定義した頂点や面に影響を及ぼさない。このため、少ない数の名称を繰り返し使用することにより、テーブル管理に要する時間を節約することができる。最適化保存では、このような処理を行っているため、保存に要する時間は若干長くなるが、次にロードする際に要する時間を大幅に軽減することができる。

### (2) ファイル・サイズ

モデルを記述する LSS-G ファイルは、以下の方法により、ファイル・サイズを節約している。

①繰り返し配置されるオブジェクトに関しては、一つのグループだけを使用し、それらの一群を定義する親グループとの間に、配置する個数だけのリンクを定義し、それぞれのリンクに位置情報を示すマトリクスを記述することで、ファイル・サイズを小さくしている。このことは反面、複数個配置されたオブジェクトを編集する（例えばカラーを変更する）場合の影響範囲をわかりにくくしている。

②パラメトリックな部品に関しては、部品名称とパラメータだけをファイル保存し、実際に表示する際に必要となる頂点や面などの情報は、ロードされた状態でメモリ上のみ展開する。このことの効果は、たとえば滑らかに見せるために多くの面に分割する曲面をもつ要素の定義において大変効果がある。

### (3) データ規模と作業量

三次元データのサイズは、しばしば面の数（ポリゴン数）によって記述される。これは表示処理に要する速度に関係している。しかしながら、データ構築に際して要する作業量（手間、従って労賃に関連）は、モデルを記述する LSS-G ファイルのデータ規模と、必ずしも直線的に相関するわけではない。例えば、曲面を多く有する部品を多数配置した場合には、ポリゴン数は急速に増大する。

これに対して、グループ数は、形状生成や配置の操作回数に概ね比例して増大するため、あるデータを作成するために要した手間を評価するためのよい指標となる。実際に、2001年度に市街地のデータを計画案に基づいて、学生アルバイトの作業により作成した際には、作業時間と、作成されたデータのグループ数に関する記録を行った。この分析に基づいて、1グループ当たりの時間、従って作業単価を設定することができた。

グループ数は、sim.exe の報告書作成機能（メイン画面のメニュー [ファイル] [報告書執筆]）により表示することができる。

#### （４）ハードウェアの進化とアルゴリズムの改善

景観シミュレータが処理対象とする、ないし処理可能なデータの規模は、現場で使われている PC のメモリ量や CPU 速度の向上に伴い、大幅に向上してきた。同一のハードウェア環境において、データ作成方法や処理アルゴリズムの工夫により速度や容量を改善する方法は試みられるが、例えば地形を構成する多くの三角形群を擬似的な一つの巨大なポリゴンとして OpenGL の表示系に送出处理する方法などは、地形編集操作も含めると複雑化につながる危険性がある。基本的なデータ構造は、単純明快な形を維持することが適切と考えた。特に速度が要求されるドライブシミュレータや、ゲームなどにおいては、見えないう側の壁面を省略した建物等を書割的に用いる場合がしばしば見られる。

2001 年度に作成した全国 15ヶ所の現場のデータ作成に際しては、主に学生アルバイトにより、市街地再開発計画等の三次元データ化の作業を行った。その際に、学生に処理能力の高いマシンを与えると、植栽等の配置により、そのマシンで表示が快適に行える限界までマニアックに複雑なデータを作成する傾向が見られた。そのような大きなデータは、一般市民が使用しているパソコンに WEB 配信しても軽快に動かない可能性が高い。このため、研究室にある旧式の、処理速度の遅いマシンでデータ作成を行わせるように改めた。このように、目的に適した「複雑さ」のデータを作成するためには、あえて処理能力を落とした環境で制作作業を行う、ということも一つの方法である。

### 16-6. 情報の持続性

景観シミュレーションは、一時的な目的で行われる場合もありうる。作業を行い、検討終了後はデータを破棄して、別の作業に移行するような用法である。しかし、一度作成したデータを、部品として再利用したり、以後の施工・維持管理に向けて継続的に活用したりすることが考えられる。しかし、現実には、そのことを阻むいくつかの障害が存在する。

#### ①物理的に永久保存することの困難

デジタルデータとして保存するためには、具体的な記憶媒体の上に保存する必要がある。メンテナンスなしで永続的な保存を保証するようなメディアは今のところ存在しない。

#### ②物的媒体の陳腐化

データのある媒体に記録しても、その媒体からデータを読み出すための装置が無くなれば、利用が困難となる。例えば、媒体そのものは完全な形で保存されていたとしても、5 インチ・フロッピーディスク媒体に保存されている CAD データを読み出そうとすると、ドライブが稀少となっている。

筆者の研究室においても、8 インチ、5 インチのフロッピーディスク約 400 枚に記録されたデータを、1999 年に CD-R に集約する作業を行った。当時 3.5 インチ FDD が主流となっていたため、中古の 5 インチドライブを、読み取り不能になる都度別のドライブに交

換しながら何とか変換した。その CD-R も 10 年を経過し、一部は読み取りエラーを生じるようになっていく。

ハードディスクに集約したデータも、最近のマザーボードに更新すると、IDE のインターフェイスが接続できない場合がある。

この困難を避けるためには、時代の流れに沿って、メディア変換を行わなければならない。つまりは、運用システムという社会的な仕組みが必要となる。

### ③データ・フォーマットの変化

データそのものが読み出し可能であったとしても、古い形式の DOS (例えば N88BASIC や CP・M) やファイル・フォーマットで記録されたデータは、処理が難しくなる。1980 年代後半に途上国で広く使われていた Word Star の文書なども、当時テキスト形式で保存していないと、利用がかなり困難である。同じ系列の CAD システム等においても、初期のバージョンのデータがそのまま利用できる保障はない。

二次元画像の分野では、JPEG 形式が広く使われている。データ形式が公開され、入出力のライブラリも、ソースコードとして自由に利用することができ、景観シミュレータにおいても活用している。この仕様を改良しようという計画も聞かれない。データ・フォーマットのあるべき姿を示す一例と言えよう。これに対して、三次元データ形式はまだ定番が成立しておらず、引き続き模索段階にあるように見える。

### ④OS の変化とアプリケーション

OS のバージョン・アップが行われると、古いアプリケーションがそのままでは正常に動作しない場合が生じる。第 1 章で論じたように、その原因の多くは潜在バグの顕在化である可能性が高いと考えられるが、ソースコードが入手できない場合には、対応することができない。商用アプリケーションの場合には、ファイルを開くことができても、グラフィックな表現において微妙に位置ずれが生じるといった障害はしばしば見られる。

アプリケーション自体が廃止となった場合には、データを処理するためには、過去のアプリケーションが動作する OS 環境を用意しなければならない。

凡そ、以上のような状況により、作成されたデータは、日常的に使い続けられるものではない場合には、いずれ使用不可能となる可能性が高い。

## 16-7. ソフトウェアの完成の条件

IT の技術開発は、当初のアイデア・構想から出発し、一通りの動作を実現する製品に到達した段階で、また外注においては、初期の仕様を満足する成果品が動作するようになった段階で、一定の完成といえる。この段階を第一の完成とすると、そこにいたる過程では、集中的な予算・マンパワーの投入が行われる。

しかしながら、実際の業務の現場に投入し、安心して利用できるツールとなることを第二の完成とするならば、そのためには、第一の完成までに投入したものと同等以上の時間とコストが必要である。この過程は必ずしも労働集約的に行われることが効果的ではない。



潜在的なバグを、一見正常に動作している環境下で、ソースコードの点検だけから発見することは容易ではない。建築材料に喩えて言えば、耐久性試験のような性格の取り組みである。OS や開発環境の変化、マシン性能の向上とデータ規模の拡大に対応することの中で、時間をかけて達成できるプロセスである。

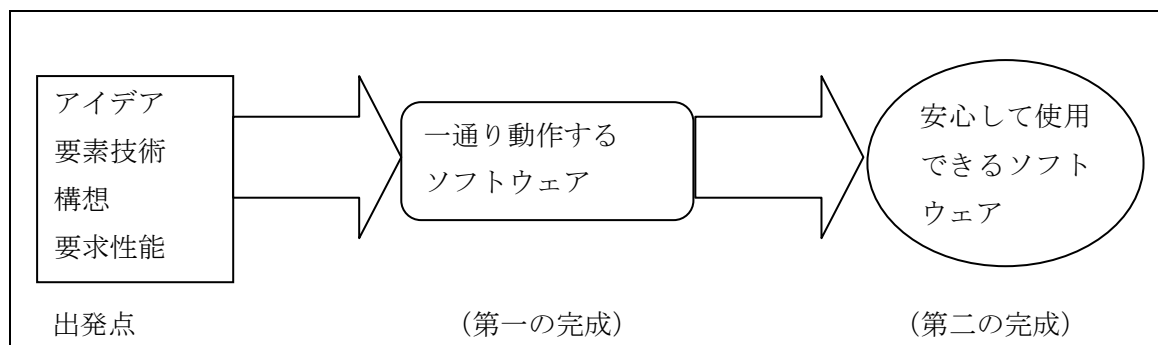


図 1 6 - 2 : システムの完成に至る二つの工程

建築工事に喩えるならば、第一の完成は上棟段階であり、大勢の職人や大量の資材が動く。以後、造作工事に入ると、現場で働く大工は少数の熟練者となり、外見上は少しずつしか工事は進んでいないように見えるが、内部では細かな神経を注いだ作業が行われている。入居後の不測の雨漏りへの対応などもこのプロセスには含まれるであろう。IT がドッグイヤーで進展する状況下では、この第二の完成に至る前に、新たな開発が開始されるような場合も多いと考えられる。

景観シミュレーション・システムの基幹部分に関しては、この後段の第二の完成に向けた具体的な作業として、以下のような事柄が進行した。

① データ処理における可逆性の確保

データの構築部分が完成すれば、当初のアイデア・構想の正しさは証明され、ソフトウェアの一通りのデモンストレーション等は可能である。しかし、削除処理などに伴い、構築したデータに伴うメモリブロックの解放等の処理において可逆性が確保されていないと、長時間・長期にわたる操作に伴い障害が生じる。

② 潜在バグの除去

スタック上の変数や解放済みのメモリ・ブロックへのポインタが使い続けられることによる障害など、直ちにエラーとはならず、原因からしばらく経過してから、時に顕在化するようなバグは、異なる OS や開発環境への載せ替えなど、長期的なメンテナンスを通じて初めて発見される場合がある。

③ 想定外のデータへの対応

三次元データには、幾何学的にありえない形状に対応したものも含め、様々なデータが生じうる。それらに対して、スキップし、エラー・ログを作成し、あるいは自動的に回復処理を行うなどの処理を実現するためには、正常なデータを通常処理する以上のステップ数を必要とする場合がある。これも、様々な用途のために実際にシステムを活用したり、ファイル・コンバーターによりインポートした様々なデータを処理したりする

ことを通じて鍛えられる部分である。とりわけ、外注により開発し、納品された状態のソフトウェアは、小さなサンプル・データによってのみ動作が確認されている場合が少なくない。発注者側において、なるべく間を措かずに、実際のデータを用いて動作テストすることもまた有効である。

#### ④ 想定外のオペレーションへの対応

システムに初めて触れるユーザーの操作は、様々でありうる。マニュアルに書かれていない様々な「行儀の悪い」操作に対しても、システム・ダウンに陥ることなく、適切なエラーメッセージを出して対応するように改良することは、幾何学的・物理的対象物に関するデータの処理というよりは、むしろ人間工学的な性格を帯びている。そして完成されたシステムを記述するソースコードのうちのかなりのステップ数は、このようなエラー・コーディングのために費やされることとなる。

当初の構想と開発目標においては、設計対象となる土木建築施設やその周辺環境の特性とモデリング方法、幾何学的演算処理、コンピュータ環境（ハード・OS・開発環境）が主たる要素であったが、実用化（第二段階の完成）に向けては、実際の運用に際しての、ヒューマン・エラーへの対応など、むしろ人間工学的な課題への対応が求められ、そのためのエラー・コーディングが必要となった。

どのように操作し、どのようにメッセージを出すかというインターフェースに関する標準化が進み、言葉や生活慣習と同様の社会的資産として共有されアプリケーション開発の標準的条件となるならば、今後同様のシステム開発を行うプログラマの負担は軽減されるであろう。

#### ⑤ OS や開発環境の変化を超えた価値

OS や開発環境の変化は、ゆるやかなマルチ・プラットフォーム対応が時系列で展開したものと見ることもできる。その中で、結果的に制作されたソースコード群の価値で普遍性の高いコアの部分は、ライブラリ関数の中にあると考えられる。一方、前項で触れたユーザーへの人間的対応に関するノウハウ等は、OS に強く依存したリソースファイルやダイアログ・ハンドラの中に蓄積されている。しかしこれらは直ちに別のシステムに載せかえることができるソフトウェア資産ではなく、その背後にある共通で変わらない、より抽象的なノウハウやロジックに価値があると見るべきだろう。

#### ⑥ 将来の拡張方法と、完成部分の分離

システムに対する機能的要求には終わりはない。とりわけ、三次元データ処理はまだ流動的な分野であり、多くのアイデアやニーズが存在している。これらに対応し続ける限り、完成は永久に訪れない。このため、Ver.2.09 においては、今後原則的に手を加えない基幹部分と、それとは様々なリスクを遮断した形で柔軟に新たな機能追加を行うためのプラグインのアーキテクチャにとりまとめた。

#### ⑦ アーカイブ化と記録・解説資料の整備

同じような開発が繰り返され、その中で同じような試行錯誤が繰り返されることは無

駄である。「最先端の技術」の名の下に、社会的に共有されるべき知識・情報が不必要に隠匿されることもまた社会的には損失である。このためには、すでに当たり前となった技術を整理し、再利用できる形で解説・公開する営みが一方で必要である。

1986年頃、建設省においてニュー・フロンティアに関する関心が高まり、建築研究所において、宇宙開発に関する勉強を少し行った時期がある。それに参加したときに聞いた、「安全性・信頼性の高い宇宙船を実現するためには、使用する部品や要素技術には、古くからの枯れたものを使う」という、当時の国際宇宙ステーション開発担当者の言葉が印象に残っている。