

# GeneXus の科学的・技術的な基礎

copyright © 1988, 2012 Breogán Gonda and Juan Nicolás Jodal

all rights reserved. 2012 年 2 月

## 要約

GeneXus は、システム開発と保守のツールだ。

GeneXus の技術によって、業務要件を知識ベース化し、システムを自動生成することが可能になる。この技術を実現するために、システム開発と保守の問題を数理論理学の原理に基づいて定義し、厳密な処理ができるようにした。

この技術の目標は、**時間が経(た)つにつれて変化する実世界を適切に表現し、常に最新の状態に保(たも)てるモデルを実現することだ。**

この目標を達成するには、実世界を適切に表現しなければならない。ここでは、ユーザーがデータに対して持つ視点を「ユーザービュー」として記述することから始める。この記述は、実世界を正しく表現する十分な能力を持つ GeneXus オブジェクトを使って実現できる。

次に、そのユーザービューに含まれる知識を抽出し、統合した後でモデルを構築する。

GeneXus は、このモデルを使った知識を格納しているデータベースからなる知識ベースと、蓄積された知識から論理的な推論により別の知識も獲得できる推論メカニズムに依存している。

この推論は、数学、述語論理、コンピュータ・サイエンスと人工知能及びツールによって行われる決定論的な演算だ。推論された知識は、ごく単純なものからデータベース設計やプログラムの生成といった複雑なものにまで及ぶ。

以上は、1984 年より我々数学者と技術者からなるチームが、数百人年の工数を掛けた研究と開発によって成し遂げられた研究成果だ。

GeneXus の最初のバージョンは、今から 20 年以上前に発表され、現在約 7,000 社の顧客と 85,000 人の開発者が GeneXus を使用している。

通常、GeneXus は大規模なミッション・クリティカル・システムの構築に使われている。

GeneXus は、『プログラミングをする代わりに業務を記述することで、システムの開発と保守を可能とするためのツール』だ。

## 1. はじめに

GeneXus を実現するための研究開発は 1984 年に遡（さかのぼ）る。

将来の変化を見据え、技術革新に対応できる製品を顧客に提供することが目標だった。当時の IT 事情は現在とは全く違ったものだった。

殆（ほとん）どのアプリケーションはバッチ処理で、対話型アプリケーションの画面は文字だけの質素なものだった。通常、当時のエンドユーザーは企業のスタッフだった。しかし現在は、世界中の重要な立場を占める多くの人々が、数多くのシステムの潜在的なユーザーなのだ。

当時、大手企業のデータベースの容量はわずか数ギガバイトだったが、現在ではヘキサバイト単位の規模になっている。つまり、100 万倍も大きくなっているのだ。

我々が考えたのは、アプリケーションの開発と保守を工業化することによって、指数関数的に増加する費用の削減と人為的ミスの発生を回避することだった。

そこで、我々が直面した単純な疑問は、何を自動化できて、何を自動化できないか、ということだった。我々は、GeneXus を使って自動化できるものは総（すべ）て自動化することを目標とした。

GeneXus というのは、高度な科学的・技術的な専門性を持つ人材が、意欲的に、信念と志を持って、様々な苦勞を乗り越えながら、やりがいを感じて仕事に取り組み、作り上げた製品だ。そして現在、GeneXus は我々の研究所で達成された成果を越えて、将来に向け GeneXus コミュニティとの連携を更に強化している。現在、世界で 85,000 人以上の技術者が GeneXus の専門家として自社のシステム全体を請け負う業務に就いている。

私たちは、GeneXus によって提供された可能性には大きな関心を示している。しかし、GeneXus がある種の魔法を見ているように思えるほど進化しているため、『彼らがこれまで見てきたどのような技術とも非常に異なっている』という事実を理由に、戦略的道具として GeneXus を採用することに同意しない、というケースにも度々遭遇した。

歴史上の総（すべ）ての偉大な革新を調べてみると、このようなケースは決して目新しいものではない。著名な SF 作家の Artur C. Clark は次のように述べている。「高度に発達した科学は、魔法と見分けが付かない」

（ <http://www.quotationspage.com/quote/776.html> ）。

実際、GeneXus の技術は魔法ではない。明確な目標と、絶え間ない努力、少しの優れたアイデアと数年に亘(わた)る絶え間ない研究開発、そして、数学、論理学、コンピュータ・サイエンスと人工知能といった利用可能な科学と技術を集結した結果なのだ。

1984 年に研究開発を始めて以来、基本的な要素を時代の要請に従ってインクリメンタル(漸増的)に発展させ、独自の理論的な発見に基づいて新しいツールを開発した。

これらの基盤のもとで業務システムの実体のメタモデルを設計し、常に改善をはかり、特定の企業に対し、特定の時期に、必要な具体的なモデルを取得することができ、その後のアプリケーション(データベース、プログラム、プロセス)の自動的生成と保守を可能にした。

## 2. 目標

そのアイデアは、通常総(すべ)でのシステム開発に影響を与えるような、多くの問題の解決を見付けることを目標としている。

- 「業務」ではなく「技術」にフォーカスしてしまう。
- 使用される「技術」のレベルが低いため、生産性が低く、人為的なミスが生じ、開発者の意欲を低下させる。
- 開発ドキュメントは、必ずと言っていいほど不十分で、改訂版もない。知識の一部は開発者の頭の中にしかない。
- 新しい技術でも、何年かが経(た)つと開発者の知識が陳腐化してしまう。

これらの問題をどう解決すれば良いのだろうか。

### 自動化できるものは総(すべ)で自動化しよう

システム開発では、何が自動化できて、何ができないのだろうか。GeneXus の 20 年以上の実績と数千の顧客の声を聞き、世界中で数万人の開発者に使用されている実績から次のことが言える。

**総(すべ)でのシステム開発と保守に関する事項は自動化できるだろう。**

- 『システム開発と保守』とは、プロジェクト、データベース、プログラムとプロセスの設計、生成と保守のことを指し、元々ルーティン・ワーク的な創造性に欠ける作業で、適切なモデルから自動的に推論できる。
- しかし、総(すべ)での会社は、その都度、異なった構造を必要とする特

定のモデルを必要としている。モデルを構築するのは、顧客のニーズを正確に理解するという創造的な作業なのだ。

### 3. 目標を達成するための要素

**モデル (Model)**: 最も肝心な点は、実世界を十分に表現するモデルを構築することだ。

**客観性 (Objectivity)**: モデルに主観的な要素を取り入れると、モデルを作成する個人の能力に大いに依存することになる。問題の大きさが増すにつれて、主観的な見方はますます重要な誤りを増加させることにつながる。モデルは客観的な要素に基づいて造られなければならない。

**自動処理 (Automatic Treatment)**: 個々のモデルは膨大な知識を集めたものだ。もし、その知識を手作業で扱うことになれば、生産性は向上せず、開発者のミスを避ける方法はない。従って、モデルに投入される知識は、人手を介在させることなく、システムが自動的に処理する。

**インкреメンタル開発 (Incremental Development)**: 実世界は静的ではないので、どのような静的なモデルであっても、時間が経(た)つと実世界を正しく反映できなくなる。だから、実世界に永久に適用できるモデルは、モデルを壊して作り直すのではなく、インクリメンタルな方式で実世界に順応可能なものでなければならない。

**IT 技術からの独立 (Independence from Computer Technology)**: 通常、常に進化を続ける IT 技術は破壊的な側面を持っている。次々と、新しい OS、DBMS、プログラミング言語、通信モニターやメタファーが登場する。つまり、バッチ処理、テキスト画面による対話処理、Windows、Web、スマートフォン、... 次に、何が現れるのだろうか。

もし、ある時期の IT 技術に基づいたモデルを設計してしまうと、次に新たに出現した技術に移行した時には陳腐化している。

以上のことから、理論的モデルは知識の正規表現であって、現在の IT 技術からは隔離されていなければならない。

### 4. モデルと知識ベース

目的は、純粋な知識を使って、格納された知識から論理的に推論可能な他(ほか)の総(すべ)ての知識を自動的に得ることだ。

このモデルは総(すべ)ての知識を集めた理論的な概念だ。我々の理論的な

モデルは、GeneXus コミュニティと知識ベースに基づく開発を進めたがっている別の会社を支援するために、2010年に公開された。

知識に基づく作業を自動化する必要性のために、以下の2つの要素で構成される知識ベースを実装した。

- **データベース**

知識が格納されるデータベースは市場から提供されるリレーショナル・データベースであり、例えば、事前に定義されていない複雑なオブジェクトや要素を対象として、意味を単純に拡張したものを集めたものだ。

- **推論メカニズム**

推論メカニズムはこの技術の基本をなすなものだ。一見、データベースに格納されたものは、昔のデータ辞典を高度化したものと感じられるかもしれない。

しかし、強力な洗練された推論メカニズムを加えたところに違いがある。データ辞書のように、保管したデータをいつでも呼び出すことができ、相互参照で関連付けができる。しかし、これだけではまだまだ不十分で、事前に格納されていない知識でも、格納されている知識の中から推論できることも必要だ。

## **基本的な例**

一連のデータビューから、要件を満たす最小のリレーショナル・モデルを取得する。

あるデータビューから、対応するプログラムを生成する。

## **5. 知識を記述する客観性と遠近法との類似性**

絵画や図は、数世紀に亘（わた）り、皆に支持されるルールがなく、観察と経験のみによって発展した。

1417年、イタリアのフィレンツェで芸術家と建築家であるフィリッポ・ブルネスキ（Filippo Brunelleschi）が、現在の建築にも適用されている記述的幾何学の原理に基づく建築のルールを定めた。

しかし、遠近法がデータ解析とどんな関係があるのだろうか？

我々は実世界を記述するための客観性を求めていた。通常、今日の各企業の現実とは異なっていて、絵を描くよりもっと複雑だが、遠近法はひらめきを生む大きなきっかけとなった。

ここで最も重要なことは、それがパラダイムシフトの基本的な原理であったことだ。絵を描く時に遠近法を取り入れたことによって、以前は複雑で、不明確な、主観的であったアプローチから（自分が「見たと思った」ことを描く）単純で客観性に基づく記述的なアプローチに変わったのだ（「見える」ものを描く）。

もし、データも同様に見ることができれば、大きな進歩を得られるだろう。実際、次のように言える。**実世界を記述するために必要な総（すべ）ての知識は、ユーザービューから得られる。**

もし、「会社の中に十分な客観性と詳細なデータを持っている人はいるだろうか？」という質問をすると、返ってくる答えは、誰（だれ）もいないという答えだろう。

だから、我々が今必要としているモデルを作るため、肯定的な答えが返ってくるように、質問を換えてみよう。

**会社のユーザーが客観的で詳細な知識を持っている話題って何だろうか？**

総（すべ）てのユーザーは、自分が処理しなければならないデータのビューについては豊富な知識を持っている。

遠近法の話に戻って、我々が行う作業でデータ分析との共通点は、「ビュー（見えるもの）の記述」をすることだ。

これらのビューから、知識が得られるのだ。

## 6. モデルの概要

誰（だれ）でも以下の疑問を持つだろう。

**ある一連のデータビューから、満足のいくデータモデルを推論可能か？**

これは、問題を数学の世界に導いてくれる良い疑問だ。

**まず、参照の枠組みを決めなければならない：**データの各要素の名称は、自動処理ができるよう、正確な命名規則に従わなければならない。

**次に、各々のビューは構造を持つ：**それを精密に、かつ客観的に表さなければならない。

そして、その一連のデータビューをモデルに移す手順が必要となる。では、上記の疑問を言い換えよう： ある一連のデータビューに対して、対応可能な最小リレーショナル・モデルは存在するか。

答えは YES だ。我々は問題を厳密に提起し、定式化した上で、工学の世界に戻って、モデルを作る手順を開発した。

要するに、問題を厳密に提起し、その問題を解決するため、数学、論理学、人工知能と工学に基づいた様々な方法とツールを採用したのだ。

## 7. 自動生成

データビューから得られた知識に基づくモデルを使って、今必要とするプログラムを自動生成できるのだろうか。

当初我々は、プログラム生成について一切考えてはいなかったが、残念なことに、周知の「安定したデータベース」を使ったシステム開発が続けて失敗していたため、遅かれ早かれ、この問題に直面しなければならないと思うようになった。そうでなければ、我々のクライアントは常に高額な保守費用の圧力に悩まされるだろう。

話題は全く新しいものではなかった： コンピュータ・サイエンスの初期から、プログラム自動生成の問題に取り組みれていた。ジェネレータは「テンプレート」か「ひな型」に基づいた空白穴埋め方式のアプローチから始まって、洗練さを増し、インストールを設定するのに必要なかなりの部分を解決できるまでになった。しかし、我々は必要なプログラム総（すべ）てを生成しなかったため、これは我々が探していたものではなかった。そうでなければ、自動的な保守機能は提供できなかつたろう。しかしながら、我々にとって、遅かれ早かれ、プログラムの自動生成は対処しなければならない基本的な問題となるのは明白だった

## 8. インクレメンタル開発

安定したデータベースは存在するか

IT 技術でよく注目されるのは、データベースの安定性だ。次の仮説について考えてみよう。

もし、企業の正確なデータベースが得られれば、そのデータベースは将来に亘（わた）って安定して稼動するに違いない。だから、その基盤を使って必要なプログラムを書くように制約を付ければ良い。

この問題についても多くの議論がなされた。

しかし、その仮説は間違っている。企業が安定したモデルを持てる唯一の状態というのは、活動が停滞しているか、活動を止めてしまった組織、ということになるからだ。

だから、安定したモデルを追求するという無駄な努力に時間を浪費するのではなく、実世界の不安定なモデルを前提として考えれば良いのだ。

### 不安定なデータベースの下で、どう対処すれば良いのだろう

これは全く新しい考え方だった。概念レベルだが、そのアイデアというのは、知識ベースはインクリメンタルなやり方で企業のニーズにダイナミックに適應でき、そして、時間の経過に伴って発生する変化に対し、最善の方法で、データベース構造とプログラムに必要な変更を自動的に加えることができる、というものだ。

我々は 20 年前に成功したが、これは当時の科学・技術の分野での大きな挑戦だった。鍵（かぎ）となる点は、知識をファイルやアクセス機構など、物理的な要素から完全に切り離して表現することだった。

## 9. IT 技術革新からの独立

IT 技術は非常にダイナミックで、無秩序な側面を持って発展してきた。特に、大企業は既存の技術を考慮することなく、ビジネス上最も効果的な方法で、次々と新しい製品を市場に送り出してきた。そして、頻繁にユーザーを大混乱に陥れるような破壊的な革新技術を出してきた。

我々の目的は、ある時点で知識ベースが利用している技術から完全に独立し、新たなジェネレータを開発するだけで、古い知識ベースから将来広く使用されることになる、新たな技術を用いたアプリケーションを生成することだった。我々の顧客は、20 年以上に亘（わた）りこの技術の恩恵を享受してい。

## 10. GeneXus

我々の研究の成果として、GeneXus という製品を開発し市場に出した。

1989 年末にリリースされた最初の PC 上で稼働する GeneXus は、IBM AS/400 用のアプリケーションを生成するものだった。この最初のバージョンは、トランザクションを使ったプログラムと単純な帳票の生成だけを目的としたものだった。1991 年には、総（すべ）ての必要なプログラムを生成し、また、自動的

にそれらを保守するために、表現力を高めた新たなオブジェクトが追加された。

それ以来、GeneXus は複数の稼働環境、新たなアーキテクチャ、新たな対話形式をサポートするように機能を拡張してきた。GeneXus のユーザーは、この過程の中で、彼らが蓄積してきた知識を使って、新たな技術を用いたプログラムを生成できる、という自信を深めた。

IT 技術は常に進化を遂げていくが、時として、企業が直（す）ぐに採用できないような、新たな革新的な技術に飛躍する。1995 年には、商用目的のインターネットが全世界にリリースされ、Web アプリケーションの開発が可能となるような、一つの飛躍が起こった。そして 2010 年には、スマートフォンやタブレット端末といったスマート・デバイスの大規模な採用によって、大きな技術革新が起きた。

GeneXus は直ちに、最も広く採用されたオペレーティング・システムのサポートを含む、Web アプリケーションとスマート・デバイスの双方をサポートできるようになった。

将来にはどんな技術が横たわっているだろうか。近い将来、Microsoft 社 と Nokia 社のジョイントベンチャーが新たなタイプのスマート・デバイスを公表するだろう。他方、中期的に起こることを予測するのは困難だ。しかし、将来出現してくるどんな技術であっても、GeneXus は新たな技術のリリースに応じて、新たなジェネレータを使った最善の方法を提供するだろう。このプロセスが我々に課せられた最も大きな使命であると確信している。

（訳文校正：ジェネクス・ジャパン株式会社）