Scientific and Technological Basis of GeneXus

Reprint 2020

Summary

GeneXus is a tool for developing and maintaining systems.

GeneXus is based on the automatic management of business systems knowledge. How is this possible? By defining the issue of system development and maintenance in terms of Math and Logic to ensure a rigorous treatment of the problem considered.

The objective is to obtain a model that adequately represents reality and can stay up to date over time. How can this be achieved? Through an accurate description of reality, by describing the different views that users have of the data. Descriptions are captured in GeneXus objects with the necessary expressive power to correctly represent that reality.

The knowledge contained in those views is then captured and synthesized to build the model.

GeneXus works with the model by means of a Knowledge Base consisting of a database where the knowledge is stored, and Inference Mechanisms that allow, starting from the stored knowledge, to obtain any other knowledge that can be logically inferred from it.

This inference is a deterministic operation where Mathematical Methods and Tools, Predicate Logic, Computer Science, and Artificial Intelligence are used. The inferred knowledge ranges from simple to very complex things such as database design or the generation of any program.

All this has been achieved through in-house research, carried out since 1984 by a team of engineers and mathematicians who have dedicated several hundred person-years of research and development to the project.

The very first version of GeneXus was released over two decades ago. Today, there are 7,000 customers and over 85,000 developers worldwide who have built their careers around GeneXus.

GeneXus is typically applied to building large mission critical systems.

In short, GeneXus is a tool for system development and maintenance, that allows **describing instead of programming**.

1. Introduction

Our research to create GeneXus began in 1984. We were looking for a product that could help our customers overcome the major changes we expected would take place over time. The situation back then was very different from the current one.

Most applications were batch-oriented, and interactive applications had very basic text screens and dialogs.

In general, end users were staff members working for the organizations. Today, the majority of the world's inhabitants are potential users of many systems.

Large corporations used databases of a few Gigabytes, whereas today's corporate databases have reached several Exabytes. That is to say, they are over a million times larger!

Our objective was to “industrialize” the development and maintenance of applications to reduce costs and avoid the human errors that were growing exponentially.

The simple question we were faced with at the start was which things can be automatized, and which cannot? Our purpose was to use GeneXus to automatize everything that can be automatized.

GeneXus is a product that resulted from the efforts of a remarkable team of highly qualified professionals, who put their scientific and technological expertise to the service of the project with outstanding generosity, enthusiasm, commitment, and faith. Regardless of the hard times encountered throughout these processes, they have always felt passionate about their work.

But we must also say that today's GeneXus and, above all, that of the future, goes far beyond what can be done in a lab. GeneXus would not be possible without the increasingly important interaction with the GeneXus Community, a group of over 85,000 developers worldwide who carry out their professional activity around it.

Sometimes we have found prospects who, although they are very interested in the possibilities offered by GeneXus, are afraid to adopt it as a strategic solution because they find it so different from what they know, and so advanced, that it seems to have magical connotations.

This aspect is not new if we consider all great innovations in history. The renowned science-fiction author Arthur C. Clark referred to this when he said, “any sufficiently advanced technology is indistinguishable from magic.”

There is nothing magical about our technology: it is the result of a clear objective, a continued commitment, a few very good ideas and many hundreds of person-years of research and development using all the help of science and technology available, such as methods and tools of Mathematics, Logic, Computer Science and Artificial Intelligence.

We started out with this toolkit in 1984 and have continually increased it with our own theoretical findings and the new tools we have built over time.

We have used this foundation to build and continuously improve a Meta-Model of the reality of Business Systems, which allows us to obtain the specific models required by each company at any given time and to automatically generate and maintain their applications (database, programs, and processes).

2. Objective

It's about finding the solution to a number of problems that commonly affect all system developments:

- The focus is on technology instead of on the business.
- The low-level technology applied leads to very low productivity, human errors and lack of motivation among developers.
- The documentation available is often incomplete and outdated, and part of the knowledge is only in the minds of some developers.
- As new technologies are introduced, the technical knowledge of those developers becomes obsolete in just a few years' time.

So, how can this situation be overcome?

By automating everything that can be automatized.

What can be automatized and what cannot be automatized in System Development? GeneXus' experience of over 20 years helping thousands of customers, and being used by thousands of developers worldwide has shown that:

The entire development and maintenance of systems can be automatized.

- By system development and maintenance, we mean **design, generation, and maintenance of the database, programs, and processes**, all of which could be ultimately considered as non-creative routine tasks, as they can be automatically inferred from a suitable model.

- Every company needs a specific model that is different in each case. Building the model implies the creative task of **understanding the needs of customers**.

3. How to Attain the Objective

Model. The first essential element to achieve the proposed objective is the design of a model that adequately represents reality.

Objectivity. The introduction of subjective elements in our model will make it largely dependent on the individuals who build it. As the size of the problem increases, the subjectivity aspect is associated with increasingly serious errors.

Our model must be built based on objective elements.

Automatic Management. Each model contains a great amount of knowledge, which does not yield significant productivity increases when used manually. Also, we will not be able to avoid the numerous human errors made by developers.

To achieve our goal, we must create a model that allows the knowledge contained in it to be used by programs, without human intervention.

Incremental Development. Reality is not static and no static model will be able to represent it accurately over time: we need the model to be able to continuously adapt to reality in an incremental, non-destructive way.

Independence from Information Technology. Information Technology is in continuous development and, in general, this evolution is disruptive.

There are frequent releases of new Operating Systems, Database Management Systems, Programming Languages, Communication Monitors and Metaphors: batch-oriented, interactive with text screens, Windows, Web, Smart Devices… What will be next?

If our model depended on the Information Technology used at a certain time, it would become obsolete when moving on to new technology.

For this reason, our theoretical model must be a **canonical representation of knowledge** that completely excludes elements of current Information Technology.

4. Model and Knowledge Base

It is about working with pure knowledge in an automatic way, and deriving from the stored knowledge all the knowledge that can be logically inferred from it. The model is a theoretical concept that contains all such knowledge. Our theoretical model was made public in 2010 as a way to assist the GeneXus Community and other companies interested in moving ahead in knowledge-based development.

We must work with knowledge automatically, and to do this we have implemented our Knowledge Base, which consists of two parts:

- **Database**

The database where knowledge is stored consists of a relational database that is selected from the market and a set of simple but important extensions that allow it, for example, to work effectively with complex objects and elements that are not defined in advance.

- **Inference Mechanisms**

Inference Mechanisms are fundamental. At first, we may think that what is stored in the database is a simple improvement of old-time Data Dictionaries.

The powerful and sophisticated Inference Mechanisms added make all the difference: as in a Data Dictionary, the elements stored in it may be retrieved at any time and associated with one another through cross-references. However, this is definitely not enough: we also need to be able to obtain any knowledge that, despite not being stored, can be inferred from the knowledge that is actually stored.

Basic examples

For a set of Data Views, obtain a suitable Minimal Relational Model.

For a Data View, generate the necessary program to work with it.

5. Objectivity to Describe Knowledge, Isomorphism with Perspective

For centuries, drawing and painting were developed empirically, without any rules to support them.

In Florence, in 1417, the Italian artist and architect Filippo Brunelleschi came up with a set of rules (principles of descriptive geometry) that we still use today.

But, in what way is Perspective related to Data Analysis?

We were looking for objectivity to describe reality. The reality of today's companies is different and, in general, much more complex than that of drawing; nevertheless, Perspective constituted a great source of inspiration.

What is most important is the beginning of its paradigm shift: with the adoption of Perspective, we went from a complex, confusing and subjective approach (we draw an object “as we think it is”) to a descriptive, simple and objective approach (we draw an object “as we see it”).

If only we could do something similar with data, we would make huge progress! Actually, we can: **all the knowledge we need to describe reality can be captured from users’ views.**

If we consider the question “Who in the organization knows the data with the necessary degree of objectivity and detail?” we will see that the answer is **NOBODY**.

Therefore, let’s try to replace it with another one that can be answered with **YES** and can help us build the model we want:

What are the topics on which the Company’s users have objective and detailed knowledge?

Every user knows very well the views of the data he or she uses.

Let's go back to Perspective; in both Perspective and Data Analysis, we “describe views.”

From these views, we obtain knowledge.

6. Overview of the Model

An obvious question is: **given a set of data views, can we infer a suitable data model for them?**

This is a very good question because it takes the problem to the field of mathematics.

First, we must define a reference framework. The names of the different data elements will follow clear naming rules, so that they can be processed automatically.

Second, each view has a structure that must be represented in an accurate and objective manner.

Third, we need a procedure to move from a group of data views to the corresponding model. We have rephrased the previous question as follows: **given a set of data views, is there a suitable Minimal Relational Model for them?**

The answer is YES! So, after having carefully defined the problem and with this certainty, we return to the field of engineering and develop a procedure to obtain the model.

In summary, by now, we have carefully defined the problem, and a wide range of methods and tools from the fields of mathematics, logic, artificial intelligence and engineering assist us in solving it.

7. Automatic Management

Once we have obtained the model from the knowledge captured through data views, can we automatically generate the necessary programs?

In the beginning, we did not think about generating programs, but the failures that IT had always suffered with the so-called “Stable Databases” convinced us that, at some point, we should address this problem; otherwise, our customers would always be burdened with high maintenance costs.

The topic was not new at all. From almost the beginning, computer science has undertaken the issue of the automatic generation of programs. Generators were based on “templates” or “skeletons” which, starting with a fill-in-the-blanks approach, then became increasingly sophisticated and ended up solving a considerable part of an installation’s needs.

This was not what we were looking for because we wanted to generate “all” the necessary programs, or otherwise we would not be able to ensure automatic maintenance. However, it was clear to us that the automatic generation of programs would become, eventually, a fundamental matter to deal with.

8. Incremental Development

Are stable databases possible?

“Stable databases” are a recurring issue in computer science. The idea was this:

If we come up with the “right database” for a specific enterprise, then that database will remain stable in the future. As a result, over time, we will simply write programs that use it.

A lot has been written about this subject.

However, the thesis is false, because the only way for an Organization to have a stable model is when it has become stagnant or has died!

Therefore, it is advisable not to waste any efforts searching for such stable models, but to get ready to work with the possible, real, unstable models.

How can we do a good job with unstable databases?

It was something completely new. The idea is to have at the conceptual level a Knowledge Base that dynamically adapts, in an incremental way, to the needs of the Company and which is able to automatically generate the necessary changes in the database structure and programs, so that everything works in an optimal way over time.

It was a huge technological / scientific challenge, which we successfully overcame 20 years ago. The key is a representation of knowledge that is completely independent of physical elements such as files and access mechanisms.

9. Independence from Information Technology

Information Technology evolves in a very dynamic and somewhat anarchical way: all the big players release their innovations trying to compete in the most commercially effective way against the others and with no regard for the past. Very frequently, disruptive innovations are introduced that create serious issues for users.

Our goal was to make our Knowledge Base independent of the technology used at a given time, so that, simply by building new Generators, we could use the old knowledge to generate applications for the new technologies that may be popular in the future. Our clients have benefitted from this feature for more than 20 years.