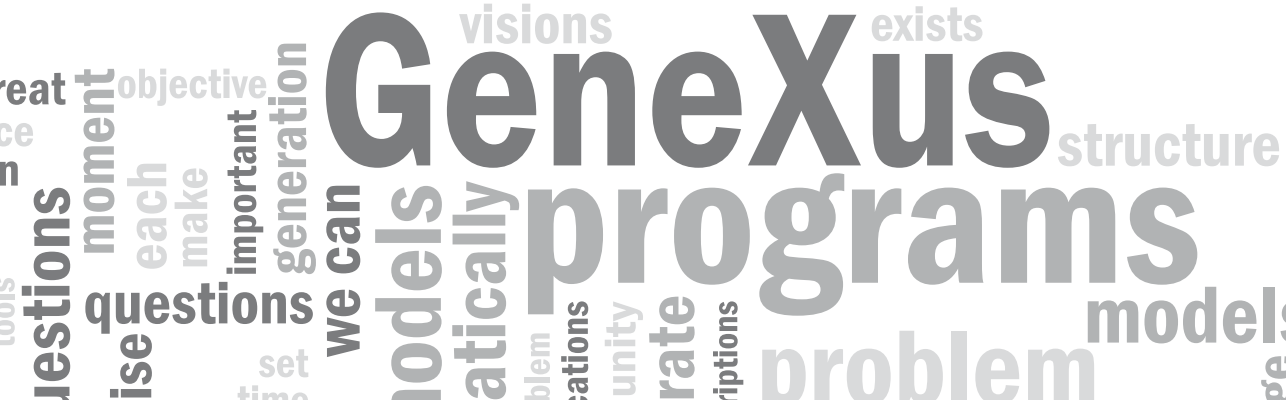


The genesis of GeneXus

Breogán Gonda | Nicolás Jodal

About the authors

Breogán Gonda

President of Artech

Systems Engineer graduated in the School of Engineering of the Universidad de la República (UdelaR).

He has been a Professor in the School of Engineering of the Universidad de la República (UdelaR), in the Pontifícia Universidade Católica de Porto Alegre (Brazil) and in the Universidad Católica del Uruguay. Within his line of studies, Mr. Gonda has delivered courses and seminars as Visiting Professor for post-graduates at universities of several Latin American countries.

From 1976 to 1989 he has provided consulting services in Database and application development projects to several of the largest Brazilian and Uruguayan companies. Mr. Gonda has delivered numerous courses within his specialty field of study in Brazil from 1976 to 1989.

He is the co-author of the GeneXus project. Together with Nicolás Jodal, his partner in the GeneXus project, Mr. Gonda was honored by the National Academy of Engineering (Uruguay) with the National Engineering Award 1995 because of their work in this project. He has been acknowledged by the Uruguayan Association of Engineers as "Outstanding Engineer of the year 1996". In July 1999, Mr. Gonda was appointed a Member of the National Engineering Academy (Uruguay).

His areas of research include: databases, artificial intelligence, automatic application development methodology and interaction between the business and information technology worlds.

He is Founding Partner and Director of Artech and GeneXus Consulting in Uruguay and GeneXus, Inc. in USA, Artech de México, Artech do Brasil and GeneXus Japan Inc. in their respective countries.

Nicolás Jodal

Vice-president of Artech

Mr. Jodal is a Systems Engineer graduated in the School of Engineering of the Universidad de la República (UdelaR).

He has been a Professor at the Universidad Católica del Uruguay. He delivered numerous courses within his specialty field of study in Brazil from 1984 to 1989.

From 1984 to 1986 he provided consulting services in database and application development projects to several of the largest Brazilian and Uruguayan companies.

He is the co-author of the GeneXus project. Together with Mr. Breogán Gonda, Mr. Jodal was distinguished by the National Academy of Engineering (Uruguay) with the National Engineering Award 1995, because of the GeneXus project.

His areas of research include: databases, artificial intelligence, automatic application development methodology and interaction between the business and information technology worlds.

Mr. Jodal is Founding Partner and Director of Artech and GeneXus Consulting in Uruguay and GeneXus, Inc. in USA, Artech de México, Artech do Brasil and GeneXus Japan Inc. in their respective countries.

The genesis of GeneXus

by Breogán Gonda and Juan Nicolás Jodal *

*We have repeatedly used the word US in this text. In certain cases this term refers to experiences or achievements of either author, or both. However, the expression mostly applies to experiences or achievements of our team. The work of that team and the generous devotion of its sometimes anonymous members constitute the greatest part of the engine that boosts GeneXus ahead.

1. Introduction

We have been requested, on numerous occasions – both from within and from outside the GeneXus Community –, to write the story of GeneXus, its origins and how we came up with it.

In our opinion, telling "how we came up with GeneXus" would not prove to be an interesting story, beyond the fact of the strong determination implied in the process, combined with endless faith, a number of small activities and some larger ones, several small discoveries and some larger ones, all of which resulted from our steady research where hundreds of man/years were consumed.

But more important than that is to mention that GeneXus is, in fact, the product of a great team of people with a high degree of scientific and technological qualifications, devotedly and faithfully working together in a generous and enthusiastic manner, always enjoying what they do, even when having to face the difficulties implied in it.

Moreover, we must also mention that GeneXus as we know it today, and most of all tomorrow's GeneXus, would not be a reality without the increasing interaction of the GeneXus Community, with its over 70,000 developers worldwide, who build their professional activity based on GeneXus.

We understand that questions are far more important than answers

As a matter of fact, we understand that questions are far more important than answers. In today's world, when questions are duly posed (rigorously formulated), we have available a huge array of tools to help us find the answer.

We are quite alone indeed when it comes to formulating those questions that lead to innovation!

However, we are quite alone indeed when it comes to formulating those questions that lead to innovation!

GeneXus is the product of a team and a Community, both of which have always been clear on the constant need for innovation, and have successfully identified the matters of significance for achieving it.

This is why we believe that re-creating certain events and questions encountered while working on GeneXus, which have proven essential in building it, will be the best way to tell GeneXus' story.

2. What did we work in before GeneXus?

A number of fields, but the ones that relate to this are a significant activity in consulting and teaching others about databases.

Our consulting activity was apparently satisfactory to clients, though not to ourselves. Most frequently, we were called upon not at the time of defining a project but when problems came up (loss of database integrity, too lengthy response times, and so on).

All clients would implement several databases, each one for treating a specific type of problem. Corporate databases did not exist.

When measured in their millions of records, those databases were considerably large, but they were quite small when considered from the viewpoint of information completeness, and specifically in relation to their usefulness for providing support to the company for decision-making.

Each client's databases were strongly redundant and consequently inconsistent. Such lack of consistency made the combination of data from the different databases a procedure not recommended even when originating in the same company.

Some of the questions that arose in such circumstances were: **Are databases actually useful? Is our activity as consultants in fact helpful?**

3.
A great project in 1984

An incidental event leading to the research that disembarked us in GeneXus was a consulting project of significance we developed by mid 1984.

An incidental event led to the research that disembarked us in GeneXus

Our client was a big Brazilian company whose head office was in the city of São Paulo. Its field of activity was mainly sports apparel and footwear. It was a large-sized company and our contact in the office was not a colleague – something quite common in those years – but the firm's General Manager. This man's thinking was absolutely clear:

"in this firm, all employees at the intermediate level, those whose decisions are not considerably significant, always have available a great degree of data support, while I, as well as other personnel in the higher management level, can never rely on the proper information to make our decisions, and our decisions are those that could make the difference between the company's success or its failure;"

"it is useless to try to define in advance the information that I will need to make my decisions. Every case is different from the others, and the information necessary can only be defined on the spot, when we actually need it: it is vital for us to define it ourselves and to obtain it immediately;"

"I have realized that we need all our systems to resort to one single corporate database that will allow us to obtain whatever data we need from it, anytime;"

"We are offering you the task of performing a full re-engineering of our computer systems in order to achieve that, where your basic workforce will be our current technical personnel, though we're also willing to hire some additional staff, in case specific qualifications or profiles, not currently available in our company, become necessary;"

"You will have state-of-the-art technology available to you at all times;"

"Still, I don't believe in lengthy projects, and we are also in an urge to have this done. It should all be ready in one year's time."

We were facing a huge challenge, as well as a great opportunity

We were facing a huge challenge, as well as a great opportunity: by then, corporate databases and corporate information were common talk, mainly in developed countries, though not much was being done in that respect.

This was the chance we had been dreaming of; we felt totally capable of taking it in and were decided to tackle it.

4. The magnitude of the problem

When having to face a real, large-scale problem like the one we encountered, where no experience has been recorded, risks become a significant issue.

When facing a new problem of considerable size, the risks involved are many

If, in addition to this, all bibliography available refers to theoretical and hypothetical cases, whose authors have not actually put hands to actually solving comparable problems, then risks and uncertainty become even greater.

But in our case, all this also increased our eagerness and appreciation of the magnitude of the opportunity we had been granted.

In the primary data analysis that was carried out during the first week of work in order to achieve a general overview of the problem we identified approximately 100 entities, and estimated that the full model would include over 500 tables (the actual model finally had 750 tables). This was a situation quite different from the usual models of the time, which never implied more than 40 tables.

First problem: How to enable end users to formulate the queries they needed to make?

Supposing we managed to build the database and the programs necessary for the application, how could we enable end users or their assistants to formulate queries as they became necessary?

This was in fact a new problem: the ideal thing would be to solve it with SQL, but then we immediately realized that such conception was not realistic. SQL is a low-level language when considered from the usability point of view, since it calls for in-depth knowledge of the database (which tables include the elements necessary and how we should navigate through them).

SQL is a too low-level language when considered from the usability point of view

We could then identify the first non-traditional problem to be solved: **how to interrogate the database at anytime, with queries that are not foreseeable, and defined by non-technical staff?**

SQL was not the solution to that, and neither were the "user-oriented languages" available at the time. These were user-friendly but in no way helped to solve the problem of large models.

The first object of our research was to obtain a language where the system would be responsible for selecting all tables necessary and for navigating through them, all of which should be done automatically.

We wanted the system to be responsible for navigating the database automatically

Data Analysis. We started by the usual detailed data analysis. Until then, we had worked with the Entity Relationship (E-R) model that dated back to the 60s, when it was introduced by Charles Bachman and later made popular by Peter Chen. We searched the organization for objects relevant to the problem and their relationships, and then represented them in the E-R model. But in this case, in principle, everything was relevant since the objective was to obtain a corporate database.

We soon realized that this procedure, quite useful in small-sized models, like those in use so far, implied several difficulties in the case of large-scale corporate models.

Also, we perceived the usefulness of visualizing parts of the model with an E-R chart. That is: the E-R model seemed as a desirable output, but a useless input in real corporate models.

Second problem: How can we build and manage large models?

Here we had data models much larger than those we were used to, and we were certainly bound to incurring in numerous technical errors derived from the large scale involved!
To aid in coping with this problem, we started to implement small tools, while we sought tools from international scenarios that could be of help to us, but unfortunately we found none.

Here we had data models much larger than those we were used to

Other problems. The difficulties derived from the model's size soon proved to be just the "tip of the iceberg."

Third problem: Where is the knowledge?

Who, in the organization, would know the data with the necessary degree of objectivity and detail? The reply was categorical: **NO ONE!**

So, what could we do? Can we solve a problem if we don't have the proper knowledge for it?: **No, we can't.**

Are there possible palliative measures like training users or scattering "section data managers" throughout the company?

Yes, that is always possible, but when we suddenly encounter problems that are 10 times larger than usual, those palliative measures are hardly sufficient. This appeared as the time for re-thinking everything with a greater degree of freedom as to admit that background information and bibliography would be minimally helpful, if helpful at all.

Where is valid knowledge located? Is it possible for us to substitute data knowledge, which we have proven to be non-existent, with other objective knowledge with the detail necessary to allow an inferred data model from it?

5.
Isomorphism with perspective

When we take a brief look at history in relation to drawing and painting as they were before, we can detect great differences with what we have in our days.

What were man's drawings like at first? How did we draw based on intuition? What are children's drawings like? All answers implied having knowledge of the object to be drawn, and better yet, the possibility of touching it, and getting to know its nature and all possible details. Then we would draw things **"as we know them."**

And what are primitive drawings like? They are significantly distorted.

And then one day the technique of "perspective" was conquered. In times of the Renaissance, certain artists – as well as architects, who encountered even more difficulties than artists in producing their drawings – started considering a change of the paradigm that said **"we draw things as we know them"** to another paradigm saying **"we draw things as we see them."**

In Florence, in 1417, in order to produce perspective drawings of buildings, the Italian art-

ist and architect Filippo Brunelleschi came up with a set of rules (principles of descriptive geometry) that we still use in our days.

Like with every new paradigm, a lot of resistance was put up in regards to perspective drawing (critics assured that "the art of drawing and painting would be replaced by a prosaic and totally uncreative technique"). However, perspective had come to stay, and in fact, it fully displaced all previous techniques.

Like with every new paradigm, a lot of resistance was put up

But, **in what way is perspective related to data analysis?**

When a decision is made to re-think everything and the thinking is done in absolute freedom, away from all possible pre-established contexts, nothing is to be discarded in advance. The story of perspective drawing constituted a great source of inspiration.

The story of perspective drawing constituted a great source of inspiration

What was most important in it was the basic principle of changing paradigms: the change was from a complex, confusing and subjective approach into another one that proved descriptive, simple and objective. If only we could do something similar with data! Then we would have a shorter way to go in pursuit of our objective.

Upon considering the problem on such bases, we should go back to the question: "Who, in the organization, would know the data with the necessary degree of objectivity and detail?" to try and replace it with another question that could be answered with a YES, which would help us in building the data model that we need.

What are the topics on which we have objective and detailed knowledge?

Both in perspective drawing and in data analysis, what we do is "to describe visions."

In our search, we found out that, just as there is lack of a good knowledge on data, every user has indeed a very good knowledge of the visions of those data he resorts to.

Going back to the issue of perspective, we could say that: both in perspective drawing and in data analysis, what we do is "to describe visions."

As promising as it seems, in perspective drawing we draw our vision directly, while here, beyond the significance of visions, what we search for is the data model.

The obvious question then is: **can we, upon a number of user visions, infer a satisfactory data model from them?**

This is a very good question because it takes the problem to the sphere of mathematics:

First of all, we must define a reference framework:

Our data elements (attributes) will be identified by name and they will basically abide by certain simple rules:

One attribute will always be identified under a single name, regardless of where it is within the model.

There will not be any two different attributes with the same name.

We shall assign names so that they best represent the meaning of each attribute.

In second place, we must represent the structure of visions:

Every vision will imply one or several attributes organized according to a specific structure.

However, it is never a good idea to reinvent the wheel:

By now, we find that mainly Jean Dominique Warnier and our friend Ken Orr on one side, and also Michael Jackson on the other, had all made great advances in describing data structures.

With their data structures, neither Warnier-Orr nor Jackson intended to define the database but rather the structure of programs, and most of the bibliography we had available referred to batch programs. But their work was a most valuable base in representing our visions of data.

The work carried out by Warnier-Orr and Jackson was a most valuable basis

Thirdly, a procedure is necessary to change from a set of data visions to the corresponding model.

We are now in the world of math, and the question is: **for a given set of data visions, is there a minimal relational model to satisfy them?**

Finally, now we have all questions relating to this matter, or better yet: we have rigorously formulated the problem.

This state allows for a great number of tools that can be of help in solving it.

In this specific case, in addition to the usual computer tools, we have also used techniques and tools taken from mathematics, logic and artificial intelligence.

We have frequently resorted to prototyping in our research work

The work done led us to solving the problem. An important byproduct was to enter a new and promising world: that of quick prototyping.

We have frequently resorted to prototyping in our research work, to then make its use viable for clients, based on GeneXus.

6.

Can we automatically generate the programs we need? All of them or just some? Which ones? And what advantages would such programs have over those written manually?

At the beginning we did not think of generating programs, but the regrettably well-known "stable databases" with which computer systems continued to fail led us to think that, sooner or later, we would have to face this issue. Otherwise, our clients would always be under the pressure of high maintenance costs.

We were no experts in the automatic generation of programs, but the topic was not new at all:

From almost the start, computer science has undertaken the issue of the automatic generation of programs.

For long, program generators were quite primitive and most of all oriented at generating simple reports based on flat files.

For long, program generators were quite primitive

After 1985 there were significant advances in this field and generators started generating both batch programs and transactional programs, and interacting with databases.

Those generators were based on templates, acting as "skeletons" which, starting with a fill-in-the-blanks approach then became increasingly sophisticated, to end up solving a reasonable part of the needs for one installation to be set up.

It was clear to us that the automatic generation of programs would become, sooner or later, a fundamental matter to deal with.

It was clear to us that the automatic generation of programs would become a fundamental matter to deal with

7. Are stable databases possible?

"Stable databases" are a recurring issue in computer science. The idea was presented this way:

If we come up with the "right database" for a specific firm, then that database will remain stable in the future. Consequently, we will limit ourselves in time to write programs that will use that base.

If that were not possible, then we have not been able to come up with the "right database."

A lot has been written about this subject.

But the basic thesis is in fact false! The only way for an organization to have a stable model is when the firm has become stagnant or has died.

Therefore, it is a good idea not to waste efforts on searching for such stable models, but rather to work on possible, real, unstable models.

8. How can we do a good job with unstable databases?

The above takes us to this question. But, if we analyze it, we can break the question down into the following questions:

- **How can we reorganize the database when it undergoes structural changes?**
- **How can we modify programs so that they can function properly with the new database?**

The first question takes us to an in-depth study of how to convert the regular contents of a database with the old structure into new contents with a new structure.

The initial theoretical problem is: **can such conversion take place without data loss?**

If the answer to this is affirmative, then further questions will come up, like: **what must we do to carry out the conversion and prevent data from being lost? and could we generate programs automatically for performing such conversion?**

The reply to the second question is obvious: if we are capable of generating the programs then why not generate all programs again?!

Resorting to brute force will almost always take us to an initial solution

Resorting to brute force will almost always take us to an initial solution, but when we are faced with thousands of programs, that doesn't seem to be the best choice, even con-

sidering the great and permanent increase in hardware power.

Surely this answer does not seem satisfactory enough, and in turn leads us to another question: upon changes to the database, **could we determine the programs that will be affected by it** in order to re-generate them?

If we manage to achieve this it will be great, but the scenario brings about yet another question: **are there programs generated for the old structure that function correctly with the new one, and are capable of being replaced by other more efficient programs?**

9. The Gods are also part of the game: "extended tables"

On a Sunday, in August 1986 in New York, when we were still far from the idea of coming up with a company and a product, and focused on putting together a number of scientific and technological discoveries under a license to be offered to others, we came upon something aesthetically fantastic: **"extended tables."**

On a Sunday, in August 1986 we came upon extended tables

We shall not go into this topic in detail here, with which the GeneXus Community is quite familiar, because the idea behind this is to focus on questions instead of on answers. But let us express the following non-essential simplified aspects:

What is an "extended table"? For each record in a specific table there is a virtual record made up of the concatenation of the original record and all records of other tables in the database that are directly or indirectly defined by it.

All those records will constitute what we call an "extended table" of the original table, and the latter is called the "base table" associated with the extended table.

How are extended tables significant?

Descriptions expressed in terms of extended tables remain in force through structural changes in the database.

Consequently, even when certain programs are actually not correct or optimal due to changes in their database, the descriptions of GeneXus user visions remain valid. And this makes it possible to automatically propagate the changes by identifying programs that are not valid to generate them again based on the original descriptions of their visions.

There was never any doubt in our minds that this discovery was aesthetically fantastic

We never had a single doubt as to this discovery being aesthetically great, and as time has passed we could prove that it is more than that.

10. GeneXus

Our intent was originally to sublicense the technology obtained to the big players in the field. But reality showed us that our endeavor was quite naive: how much credibility could such advanced ideas have for the huge technological giants in developed countries, when the ideas originate in a territory with no tradition in the production of technology?

How much credibility could such advanced ideas have when they originate in a country with no tradition in the production of technology?

The dilemma was between making our discovery public and consider the research as concluded, and putting together a company and a product to take a shot at marketing it, starting in Uruguay and the region, to then move on to other destinations worldwide.

We went for the second option. We thought that, over time, if our product was actually useful to clients and clients increased constantly – even at a slow pace –, and if we managed to duly support new technologies as they appear, we would attain success to the extent that we abode by a spartan entrepreneurial behavior.

So we were profoundly committed to such purpose, and by late 1988 we founded Artech, we named our product "GeneXus," and we set out to launch its first version in the second half of 1989.

Here are some of the questions that we came upon back then:

What are the platforms for which GeneXus must generate applications at the start?

The realistic attitude was to initially choose a single platform and then specialize on it until we could come up with a minimal volume of business that would empower us to undertake new platforms. Our choice was IBM AS/400.

In which platform must GeneXus function at the start?

What we needed was an efficient platform that would not call for significant investment, to ensure constant development.

We also sought the greatest independence possible from the platform for which we would be generating applications, so that we could then make the generation for other platforms something viable.

We also sought the greatest independence possible from the platform for which we would be generating applications

Our decision was to go with PCs with DOS operating system.

What are the problems that GeneXus must solve on the spot, and which could be postponed for later?

It was clear to us that, at the beginning, we would have to face lack of theory and practical inconveniences that would prevent us from generating full applications.

One of our objectives was to generate as much as we could, automatically.

Our second objective in mind was to automatically maintain everything we generated

Our second objective in mind was to automatically maintain everything we generated.

And this was the reason for totally discarding the generation of parts of programs for developers to take it from there, because what developers would write manually would not be automatically maintained.

We did not know how to describe procedural programs like batch processes or casuistical routines that could not be derived from transactions.

There was the chance of introducing a 4th generation language to solve all those problems, but that would imply giving up the automatic maintenance of these programs, so we decided not to go for that.

We opted for implementing that which could be automatically generated and maintained, with no restrictions, like transactions, queries and simple reports.

We estimate that the first version of GeneXus automatically generated and maintained 70% of the programs

We calculated GeneXus to automatically generate and maintain 70% of the programs in an installation, while doing the automatic design, generation and maintenance of the database. The rest would require long hours of additional research.

Could we automatically generate and maintain 100% of applications?

That was what we aimed at, and we were working on it, and relied on having quite some time to achieve it.

We believed that clients chose GeneXus because of the significant increase in development productivity that it implied

Generating and maintaining 70% of programs automatically seemed a great degree of success. There were only a few generators capable of coming up with similar things, but none of them could provide automatic maintenance.

One thing we had no doubts about was the fact that automatic maintenance was a very important and unique feature of GeneXus. But we believed that clients chose GeneXus because of the significant increase in development productivity that it implied, while automatic maintenance was something seen as an added, unexpected new feature towards which most of Artech outsiders were skeptical to a certain degree.

But reality came as a surprise when several clients brought their thoughts to us:

"To automatically generate 70% of programs will be of great help and that is something we appreciate a lot."

"Having to manually write 30% of programs is a restriction we accept in the belief that you will surely overcome this in the future."

"But what is not acceptable to us is having to do the manual maintenance of programs not generated by GeneXus."

How could we satisfy our clients? How could we be sure of generating full applications?: by means of a procedural language.

Our question was: can we build a procedural language with descriptions (source programs in that language) that will not become invalid in the event of structural modifications to the database?

The answer was YES, and allowed us to release the first full GeneXus.

Then questions started to pour in, endlessly…

Can we generate applications for other platforms and architectures?

Can we identify patterns in our descriptions?

And based on that, can we automatically generate GeneXus objects to fit those descriptions?

Can we extend GeneXus beyond Artech? Can we enable the GeneXus Community to make this possible?

… and many, many more….

Formulating new questions is a constant occupation, where there are an increasing number of people taking part of it

Formulating new questions is a constant occupation, where there are an increasing number of people taking part of it, like the research and development team, all of Artech, software houses, clients and the whole GeneXus Community.

11. Answers

Throughout all these years working, we have come upon endless questions, most of which generated the corresponding answers, a lot of answers. Nevertheless, we could summarize them all into one main principle:

We can "describe" instead of "program"!

and one strong belief:

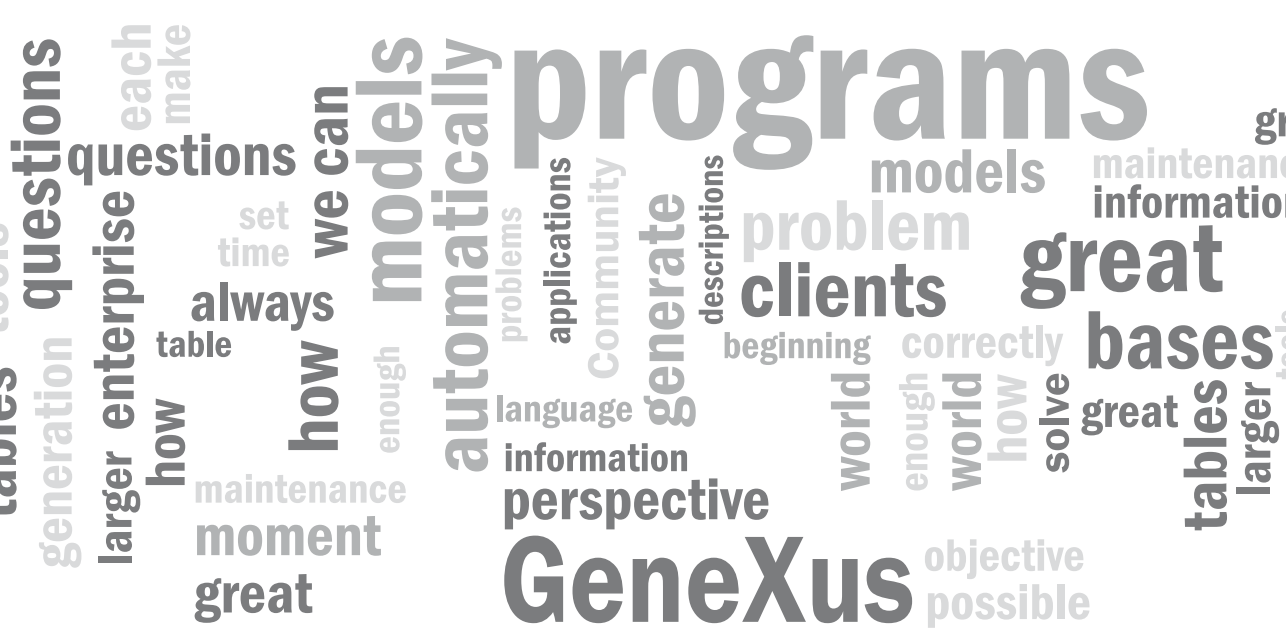
We must never lose our freedom of thought!

12. Acknowledgements

We must particularly thank a number of individuals and companies: our team, our clients, and the whole GeneXus Community.

And also, we would like to humbly express our thankfulness to those who, despite not being clients or participants in our Community, even somewhat skeptical about GeneXus, permanently bring up new difficulties and challenges, for these are our inspiration for further learning.

We have a wonderful world, and anything capable of nourishing our knowledge of it must be seen as a privilege and a unique opportunity!



Artech™
www.genexus.com

MONTEVIDEO - URUGUAY	Av. Italia 6201 - Edif. Los Pinos, P1	(598) 2601 2082
	Ruta 8 Km. 17.500 - Edif. @3 Of. 205	(598) 2518 2870
CHICAGO - USA	400 N. Michigan Ave. Suite 1600	(1 312) 836 9152
SÃO PAULO - BRASIL	Rua Samuel Morse 120 Conj. 141	(55 11) 5502 6722
CIUDAD DE MÉXICO - MÉXICO	Calle Leibnitz N° 20, desp. 801	(52 55) 5255 4733
TOKYO - JAPAN	Tk Gotanda Bldg. 303-2	(81 3) 5793 5481