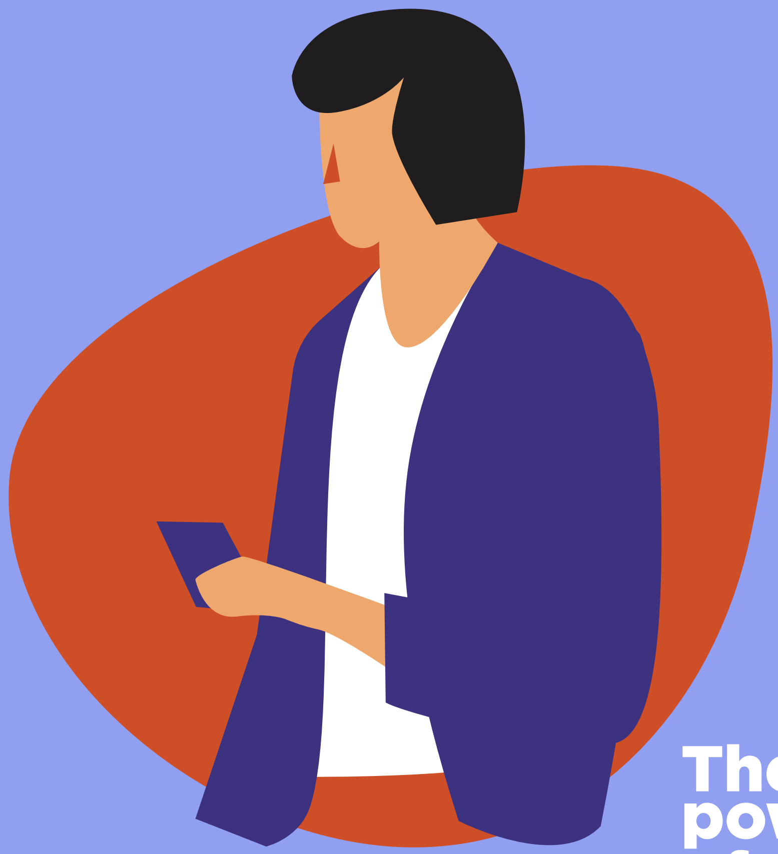
A few considerations about the genesis of GeneXus

By Breogán Gonda *

Whitepaper



The power of doing.

A few considerations about the genesis of GeneXus

By Breogán Gonda *

In 1984, a large company in Brazil, commissioned us to fully re-engineer their IT. The client wanted to develop all their systems interacting with a single corporate database.

The challenge was great: at that time, everyone talked about systems and corporate databases, but the reality was very different and all companies continued to use multiple "subject databases". Each subject database was used to support a small family of applications. Each subject database was updated with total independence from the others. As a consequence the general consistency was impossible. The companies knew this and, therefore, did not combine data from different subject databases.

In short: there was an operational computing system that operated under certain restrictive conditions, but there was no corporate computing.

The client presented us with a big challenge: a single centralized database to serve all his needs.

The client thought, with good judgment, that any information that was necessary at any time could be obtained from that corporate database.

There did not seem to be any restrictions regarding the necessary resources, but there was a deadline within one year.

The task tested our convictions, our experience, our methodologies and confronted us with a set of additional problems. All the obstacles were overcomed and the project ended successfully.

Having faced so many difficulties made us learn many things and encouraged us to continue. The question was, doing more similar projects or building our own methodology and a tool to support it? The latter was the chosen path.

We participated in several projects for different companies, while, we continued researching and, finally, we launched our first GeneXus in 1989.

Today, 34 years after that project, we come up with a set of reflections on an event that was important for our technological and business development.

We had an important contract to take a big step forward, we were a very good team, and despite many people around the world spoke or wrote a lot about corporate computing, there were no, or at least very few, successful accomplishments in the area.

We addressed the issue with great seriousness and responsibility, and what led us to face a set of problems:

ISSUES

Database. In supposing that we would attain a proper solution for the problem, our initial estimate of the database required indicated the need for over 500 tables. In the end, our consultancy work amounted to 750, and we wondered if our methods would work to solve a problem of such magnitude. For that purpose, we needed to find answers for a number of important issues, namely:

Exploiting the solution. Our client aspired to the capability, of their managerial positions, and/or their assistants, for solving their queries at any time –in most cases–, without the aid of computer experts.

The language to apply would be SQL, where for most queries, instructions become necessary in relation to which table data should be obtained from, and how such tables should be combined with one another.

Building the solution. How should we go about designing such a huge database? We were specialists, and as university professors we were confident about our good knowledge on the subject. In that regard, we were also advisors for a number of large Brazilian entities from the oil, financial, industrial, government and business fields. However, none of them aspired to large-scale corporate databases, nor to highly complex databases requiring significant transactional activity. They all had several subject databases, each of which helped in solving a small family of problems.

Data model size. And for the design stage, how should we go about reasoning with a data model of over 500 tables? How could we possibly visualize that? Could we possibly normalize it, or should we give up on that idea?

We were surely headed towards encountering a number of human errors, so, having tools to aid us with that would be highly beneficial, <u>but no such tools existed!</u>

Sources of knowledge. In the event of attaining or building the tools required, we would also need the raw materials for putting them to work. Our raw material was, at every level and very rigorously, knowledge of the user's reality (user views, business rules, etc.).

But, in order to build the ER (Entity Relationship) model we aimed at, whom did we have within the client company with the necessary knowledge about the various objects from reality and the relationships amongst them? Who possessed the objectivity and was able to provide the detail required?... simply NO ONE.

> Having tools to aid us with those problems would be highly beneficial, but no such tools existed!

System maintenance. Given the scale of the solution, and in the assumption that it was built in strict accordance with an appropriate development method, what would happen once the consultants' work was finally concluded? Would the methodology continue to be scrupulously observed? Upon resorting to documentation, would the data obtained be reliable, accurate and up to date?

The continued and precise application through time of a method used for building a solution is almost impossible, save when that solution is supported by a ***development tool*** acting as a guide that ensures strict compliance with the corresponding rules.

SOLUTIONS

Accurate and rigorous descriptions. The answer to the problem with a solution that would remain valid over time called on us to describe our client's reality in an absolutely accurate and rigorous manner.

Sources of knowledge. We concluded that the basic source of knowledge are the views of users data expressed within a solid reference framework that must also be simple

GeneXus infers the data model, the database schema, and the application programs.

Artificial Intelligence. Manual solutions could not solve such a problem, so we decided to resort to Artificial Intelligence.

Thinking that this would just imply using certain tools is an incorrect viewpoint that proves glib.

Of course, we subjected the two most widely used languages of the time to analysis. The consideration of LISP and PROLOG led to two different positions: using LISP because a plurality of language processors were available for it and it was also the most widely used and oldest language, or otherwise going for PROLOG, which was a new language with higher expectations but not so widely used and with very few language processors available. We opted for PROLOG and it was indeed a good decision.

Artificial Intelligence appeared as a very promising technology. We addressed the issue with great seriousness and responsibility, and what led us to face a set of problems: missing opportunity at a time when numerous companies, particularly in the U.S.A., were working on the development of expert systems, especially for diagnostic purposes. In diagnosing, reliability averages of, for example, 95 to 97% are excellent values (the percentage, in the case of very good human specialists, is usually lower). However, in the building of large data models and systems those figures are disastrous. A full 100% is necessary!

And there was one more thing: we were computer systems engineers with a solid mathematical background, but we knew nothing about Artificial Intelligence.

A NEW WORLD

Our problem led us to a whole new world, the one of pure knowledge!

Learning by doing. In cases where we seem unable to solve a problem but we continue studying it and finally find a solution, we actually achieve a lot more in ***learning by doing*.** Today, that is the way to keep up to date with and apply new technology immediately, with very significant benefits.

Accuracy. Working with pure knowledge suggests a number of requisites that could be summed up into a single term, and that is ***accuracy.***

All of us (researchers who were in some way part of the conception and building of GeneXus) needed to –and still do – apply a very high degree of abstraction to our activity so as to express each problem rigorously, in the form of a problem of pure mathematics and logic.

Every specification we define must be fully accurate and based on a clear and solid reference framework:

- representing the ***meaning*** for each different element involved so as to be able to operate automatically with it,
- adopting a ***basic rule*** –given the fact that reality is consistent, then all our representations of reality must also be consistent, and
- implementing ***powerful operators*** to be used automatically.
- And what about our users?... the developers. Expecting to have them constantly work with an extremely high degree of abstraction would not be realistic. We must ***hide complexity*** in allowing developers to work at a high level in a direct manner and, as simply as possible, with specific elements, using low abstraction.

Pure knowledge. Assuming pure knowledge as the baseline implies refrainment from resorting to physical or technological elements of any kind that might be subject to changes (something that will inevitably happen with the passing of time).

Physical and/or technological elements necessary are included automatically and only upon code generation.

This behavior enables a high degree of abstraction that is completely independent from casuistical variable elements. All specifications are stored at the same high level.

Futureproofing. Knowledge is stored to be operated with on a Knowledge Base with a great capacity for inference.

GeneXus infers the data model, the database schema, and the application programs.

Among the most important outcomes is the fact that, since such knowledge is independent from the technology applied, we may always automatically generate a system using another technological setup. For each case, we must simply use the GeneXus version that provides support for the technological configuration selected.

That is to say that, Futureproofing comes into action because, with GeneXus, systems become protected concerning possible changes caused by the everchanging development of technology, and ***nothing has to be reprogrammed manually!***

Automatic maintenance and evolution. Another consequence somewhat less spectacular, though equally important, is the automatic maintenance of systems. GeneXus' capability for inference originally enables the automatic generation of systems (database and programs), and in the event of changes, it propagates them automatically as well.

Developers only need to perform the conceptual update of the descriptive elements that have undergone changes. GeneXus will provide a report on the impact of such changes, and for the case of these being approved, it will automatically propagate them as well.

INTEGRATION

These days, and regardless of scale and of the field of business, no company in the world may operate on a self-sufficient basis. It has become increasingly necessary to resort to third-party discoveries, experience and products by way of supplement.

It is advisable to:

- facilitate maximum integration with other products in order to solve the actual problems affecting clients in the most comprehensive, fast and economic manner possible, and
- avoid developing what is liable of being purchased, considering the quality of the solution to be included, time for becoming available, and opportunity cost.

GeneXus' capability for integration has been significantly enhanced in its latest versions.

SUMMARY AND FUTURE

We separate knowledge management from technology management. All that's been said about our technology, and particularly about the automatic use of pure knowledge could be summed up in a statement:

"We made possible a very good automatic management of knowledge about Business Systems".

But as the world evolves, where is the limit for such Business Systems now? These days, companies that originally dealt with a number of different activities have decided to work as software companies and specialize in all those areas.

GeneXus represents knowledge in a rigorous and permanent manner that allows it to operate with knowledge automatically. What kind of knowledge? anyone! we must work hard, but there are no limits!

And that provides an open door to evolve a lot: our clients will need it. This need will turn into a must, as the continuous addition of new types of applications –unthinkable of until a short time ago– makes computer systems increasingly complex. It will take some time and big efforts to continuously evolve our intelligent development platform to do it.

The solid scientific and technical basis of GeneXus will allow us to incrementally continue building whatever becomes necessary.

***Breogán Gonda.** *Engineer, software developer, consultant, researcher, professor. At present he is Chairman of the Board of GeneXus.*

The solid scientific and technical basis of GeneXus will allow us to incrementally continue building whatever becomes necessary.

Find out how GeneXus
can do the same for your company.

www.genexus.com

GeneXus™

MONTEVIDEO - URUGUAY	Av. Italia 6201- Edif. Los Pinos, P1	(598) 2601 2082
CIUDAD DE MÉXICO - MÉXICO	Hegel N° 221, Piso 2, Polanco V Secc.	(52) 55 5255 4733
MIAMI - USA	7300 N Kendall Drive, Suite 470	(1) 201 603 2022
SÃO PAULO - BRASIL	Rua Samuel Morse 120 Conj. 141	(55) 11 4858 0300
TOKYO - JAPAN	2-27-3, Nishi-Gotanda	(81) 3 6303 9381
	Shinagawa-ku, Tokyo, 141-0031	(81) 3 6303 9980