Evolution of Genexus Objectives

Toward the Fourth Dimension

Prepared by Breogán Gonda and Juan Nicolás Jodal [1]
Artech, February 2009

What can we say, 15 years later, of Genexus objectives? [2]: How have they evolved? How was their compliance?

Genexus objectives have constantly and always evolved toward more demanding ones, in as much as our research, our general technological development and a better understanding of reality have taken us.

The facts show a high level of objective compliance.

The Four Dimensions

We've been able to identify the objectives and assess their compliance by introducing the following four dimensions:

1. **Completion** (It will be 100% if the database and all the application programs are generated by Genexus)
2. **Productivity** (It is the potential increase of productivity through a good use of Genexus, in comparison to what good developers will achieve when manually using common programming languages (such as COBOL and RPG previously or Java and C# currently)
3. **Universality** (It will be 100% if applications could be developed for any relevant platform)
4. **Usability** (It quantifies how easy it is to use Genexus: It will be 100% if anybody can use it without specific training)

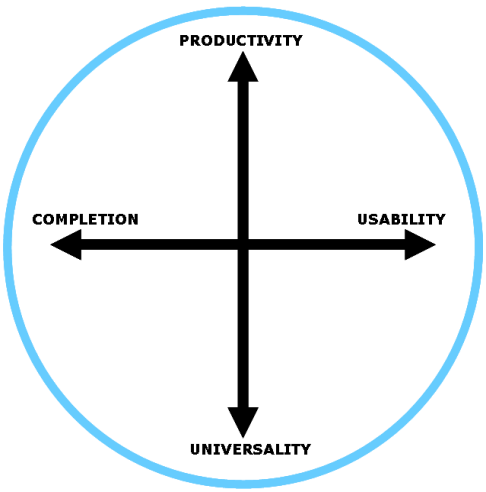
It is interesting to observe how the objectives and achievements have evolved since the first Genexus version was released at the end of 1989.

It is always difficult to represent four dimensions on paper. The following is a reasonable and descriptive representation (although not rigorous).

We represent each dimension with an arrow according to the following diagram, and we attribute to each arrow the value assigned by Genexus assessment according to that particular dimension.

We arbitrarily placed the dimensions in the following manner: **Completion** points to the left, **Productivity** points upward, **Universality** points down and **Usability** points to the right.

THE FOUR DIMENSIONS



Objectives at the Time of the Release of the First Genexus Version

When the first Genexus version was released, it already had a significant achievement in terms of Completion, Productivity and Usability.

Completion:

For the first version, our objective was to generate 70% of the programs. For each program, a 100% of the code was generated. In this way we avoided the need to manually modify the generated programs.

This characteristic was always considered critical: Since the system has full knowledge to generate both the database and the programs (those that fall within the 70% of programs that Genexus was able to generate), it also has the knowledge required to automatically maintain everything generated (Database – structure and content – and programs).

It was clear, however, that the ideal situation would be to generate a 100% of all programs. If we were able to do so, Genexus would be able to automatically maintain the whole system, causing a dramatic decrease in cost (both in time and money). The technology available at that time did not allow it.

Productivity

Productivity significantly increased since the developer did not need to dedicate time to a set of traditional tasks: Data Analysis, Database Design, Program Design and Coding.

The basic system tests are to verify the correction of specifications via live and complete prototypes, which are friendly and timely.

The objective from the start was to have a potential 500% increase of productivity compared to manual programming in the available languages (in the case of RPG and COBOL).

Usability

Usability increased in comparison to manual programming (in that first version for IBM OS/400 operating system, the Database Managing System native to AS/400, its command language and programming language that could be RPG or COBOL).

Why did usability increase? Because the developer did not need detailed knowledge of any of those elements. It then allowed the developer to be free from those low level elements and dedicate time to understand and conceptually solve the client's problem: Working to solve real problems and not problems face due to limitations in the technology being used.

It was not easy to quantify the usability increase, but it was important since it allowed both old users, already accustomed to an obsolete technology, as well as new users with little or no experience, to immediately use the new technology.

The objectives were met. What was the clients' reaction? ·Why did they adopt Genexus? What traits were most valued?

The clients' main reasons for hiring Genexus were:

- Genexus visualized application development for the new IBM AS/400 to technicians with little knowledge of this new computer and its technology.
- Genexus offered a great increase in the developers' level of productivity.
- **In particular, the fact that Genexus offered automated maintenance of all programs generated was not a critical advantage to potential clients.** Nobody really assessed this advantage seriously because no one thought automated maintenance was at all possible!

After using Genexus for a short while, the clients kept their positive prior assessment: Technicians with little AS/400 knowledge could work with no problems and great productivity.

However, the major change observed was that clients began to consider that automated maintenance of all Genexus products was a critical advantage.

At the same time, Genexus could only generate 70% of all necessary programs. This fact, considered reasonable at the time of adopting Genexus, was now perceived by customers as a great limitation. Why? Because now several Genexus advantages were becoming obvious:

- Strong increase of productivity of automated vs. manual programming.

- Great advantage of automated vs. manual maintenance.
- Automated maintenance could only be performed on automatically generated code.

As a result, users wanted to avoid the manual programming of the remaining 30% of the programs. But, more importantly, they wanted to avoid the need to manually maintain –for ever– all programs that were manually developed.

As a result of the above, two important things took place:

- Everybody understood quite well that they never had to manually modify the generated programs to preserve the automated maintenance capability.
- The clients applied a great amount of pressure on Artech to have Genexus generate a 100% of the programs.

Generating 100% of the programs was a very good objective, for many reasons. Some of those reasons were very clear:

- Automated maintenance of 100% of the application. Increase productivity as a result.
- Other reasons became clear years later.
- Application portability between platforms.
- Knowledge Base marketing, etc.

But the objective to generate and automatically maintain the whole application was not an easy objective to achieve (no other tool was doing it at that time - nor does it today – in the world). Could it be done?

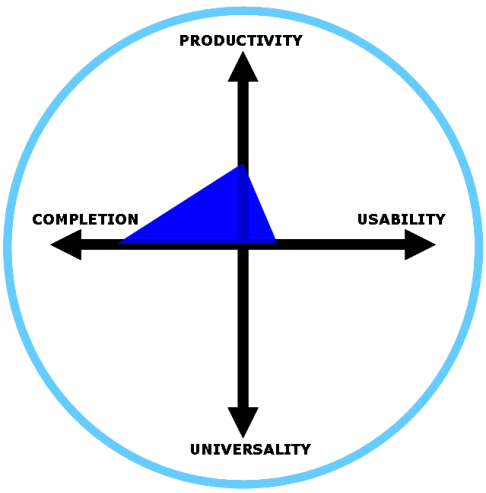
To generate 100% of an application, it is first necessary to describe it fully and accurately. At that time, Genexus did not have enough expressive power to do so.

How could Genexus add expressive power fast? Genexus was 100% declarative, if a programming language was added - such as 4th generation language - it would gain expressive power. But, at the same time, it would lose its capability to automatically maintain all programs generated, because in the procedural languages known at that time the source programs did not stay valid if the Database was modified.

It was needed a procedural language with source programs that would stay valid when the database suffered modifications (and of the highest possible level).

An immediate solution was not found, but the issue was the top priority in our research list.

OBJECTIVES AT THE TIME OF THE FIRST GENEXUS VERSION (1990)

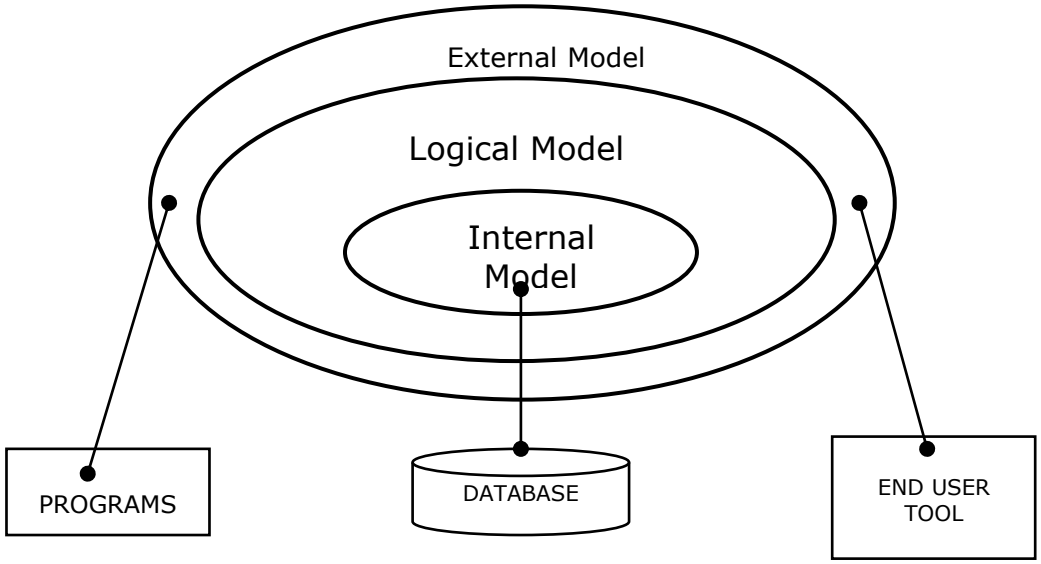


Objectives for 1992: 100% of Completion

After considering and discarding many options, we arrived at a simple approach of the problem.

A Solution through a Database Managing System. The most reasonable option (independent from Genexus) was implementing the ANSI Sparc recommendation of the 3 Model Design (External Model, Logical Model and Internal Model) by the manufacturers of Database Managing Systems. All applications would interact with the External Model, which was not going to be altered by modifications on the database.

ANSI Sparc Recommendation



The trends at that time made this option highly unlikely: The Database Managing Systems market, where several companies had participated with different products constantly competing in an innovative and enthusiastic atmosphere, adopted SQL as its standard in a version not in the least innovative.

This standard implied an almost full halt of the functionalities and the survival of only a few a those manufacturers.

In conclusion: There was no solution (and still there is none) that way.

"Stable Databases." Many theorists proposed "stable and well-designed databases in advance," and there was much talk about them (Clearly, if the database is stable, the problem we were trying to resolve would not exist: Since there are no structural modifications to the database, there is no impact on the programs).

However, these concepts are not consistent with reality: There can only be Stable Databases in declining business or organizations that have lost all their innovative capabilities.

In conclusion: It is an approach with no contact with practical reality.

Procedural Language with Programs Independent from the Database Structure. The third option was to design and develop a procedural language so that the validity of its programs would not be affected by database changes.

We adopted this approach, one that was very difficult or nearly impossible if taken in isolation, but was totally feasible in a knowledge-based environment as Genexus. Why should we manually put elements in programs (table and file names, etc.) that could be automatically inferred (at the right time, which in this case at the time of the program generation)?

Genexus procedural language acts only on the External Model (whose elements are not affected by Database changes) and does not use physical low level elements such as tables, files, etc., and its source programs are not affected by structural changes made to the Database.

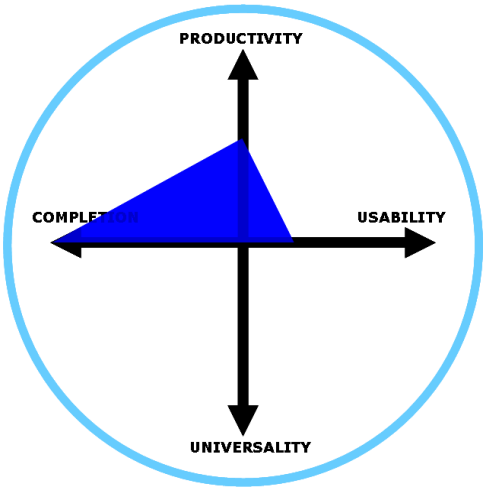
The addition of this high level procedural language allowed us to solve 100% of the problem with Genexus.

Now Genexus was capable of automatically generating and maintaining the database (structure and contents) and 100% of the programs.

This way the 100% Completion objective was achieved and we were committed to maintain it in the future. This dramatically increased the advantages of using Genexus. The users understood it.

Genexus was a tool that basically generated applications for only one platform: IBM AS/400 computer. However, there were no theoretical limitations to generalize it to other platforms.

OBJECTIVES FOR 1992: 100% OF COMPLETION



Objectives for 1995: Client/Server Architecture Support

In 1995, something long-awaited and postponed finally happened: A strong expansion of the Client / Server Architecture took place.

Artech released the Client / Server generators for the most relevant Database Managing Systems of that time: IBM DB2, IBM DB2 for AS/400, Informix, Microsoft SQL Server and Oracle.

These new capabilities had a very positive reception and the client / user segment of the Client / Server architecture started to show the fastest growth.

At the same time, the sudden and somehow unexpected release of Internet for business purposes represented a great success and the whole computer sector began to change rapidly.

Until then, the systems were foreseeable and structured, and targeted to a few million users throughout the world. These users utilized them without any degree of freedom, and only after being specially trained to do so.

The developers - a relatively small group in the world – would keep all the decisions and freedom for themselves.

In the midst of this reality, Internet turns up and offers, from the beginning, access to a much larger number of people from all over the world, people who could not be trained and with a greater degree of freedom. Genexus quickly released its first Web Generator.

What was the status by the end of 1995?

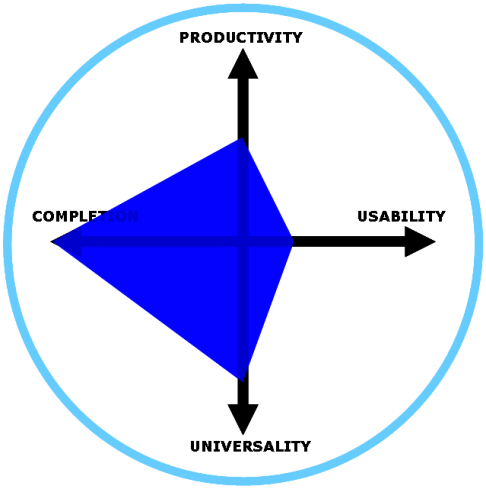
Genexus kept its **Completion**: 100% of the programs were automatically generated and maintained.

Productivity was still very high (potentially 500% higher than the productivity obtained through manual programming).

Universality substantially increased by incorporating the Client / Server and Web architectures. What was beyond Genexus reach? Basically, the mainframe applications (but a declining tendency in the development of new applications for mainframes was already clear).

Usability was kept at the previous levels.

OBJECTIVES FOR 1995 - CLIENT / SERVER ARCHITECTURE SUPPORT



Objectives for 2001: Dramatic Increase in Universality

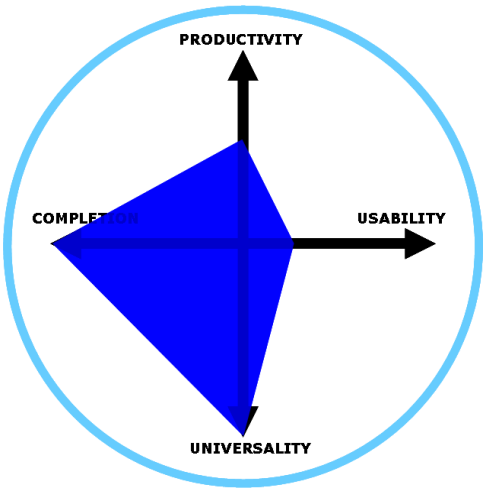
During the second half of the 90's, the concept of a web-oriented “platform” for the execution of multi-tier applications came to being; it was something the users were asking for: An automated updating of program version for PC clients.

The two rival platforms, Java and .NET, quickly split a large share of the market (and with their constant competition made it grow). Java and .NET shared the market basically with Client / Server architectures, while the Web architecture gradually started to grow, and the mainframes’ participation significantly decreased.

Genexus timely released his generators for Java and .NET, and at the same time it strongly refined its generators for the Web architecture.

Based on this, it maintained good levels of **Completion**, **Productivity** and **Usability**, and substantially increased its **Universality**: Now Genexus generated business applications for all the platforms for which new applications were really being developed.

OBJECTIVES FOR 2001 - DRAMATIC INCREASE IN UNIVERSALITY



Objectives for 2004: Dramatic Increase in Productivity

Computers have greatly changed since the release of Internet: There are many more users (actually hundreds of millions of users). **In general, these users cannot be trained.**

Databases are not just physical databases found inside a company, but Extended Databases that involve clients, suppliers, public or private Web services, etc. At the same time, our databases are accessed by other people, who are duly authorized, but in many different ways.

The devices that act as terminals have diversified, adding Hand Helds, Palms, Pocket PCs, cellular telephones, etc.

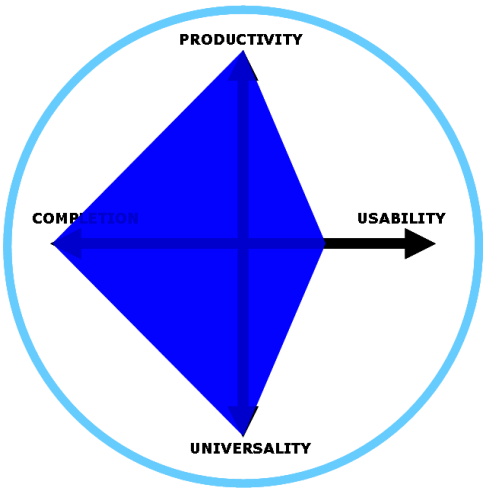
The networks have extended, now including a much higher rate of speed, wireless long-distance connections (Wi Fi), wireless intermediate-distance connections (Wi Max), high speed telephone connections, associated to cellular telephone networks (GPRS, EDGE).

However, the crucial aspect is that our applications must serve several million potential users, who have a much greater degree of freedom and who cannot be trained. As a result, the applications are more complex and need to be better designed and built to hide all that complexity and offer simple, easy to use, and intuitive interfaces to the final user (since they are generally targeted at non-cataloged, and therefore non-trainable, users).

Faced with this new reality, Artech realized that a 500% increase in potential productivity compared to manual development with a low level common language (such as Java or C#, for instance) will not be enough in the near future. Therefore, Artech replaced the traditional 500% with 2000%, which is achieved through two 100% productivity increases, each one over the immediately previous version. The first 100% comes with the version release at the end of 2005 and the second 100% will come with the version to be released in mid 2007.

At the same time, the other dimensions were carefully monitored, particularly the Universality, adding support to different platforms or platform elements that were developed and became relevant in the market, such as PostgreSQL, MySQL and AJAX.

OBJECTIVES FOR 2004: DRAMATIC INCREASE IN PRODUCTIVITY



The Future: Dramatic Increase in Usability

And now? What will be the next innovation? What can we expect from Genexus in the future?

Genexus has managed with great competence its three first dimensions:

The **Completion** and **Universality** dimensions have reached their peak level and we will continue to work to face the technological innovations that will come in the near future.

Productivity: The productivity increase is such that Genexus will be able to develop- timely and with reasonable costs - increasingly complex applications that the users will be demanding in the next few years. This will be of the utmost importance since it won't be possible to do it manually with common programming languages such as Java and C#.

In short, what is the status of the Fourth Dimension (Usability)?

Usability is good, but we cannot compare it to the level of the other dimensions. Genexus is a tool for developers with a sold algorithmic foundation, for example, since its procedural component is still important. Its use must become much friendlier to substantially increase its Usability, so that anybody with a solid general background could use it while performing normal tasks without having to go through an expensive and specific training.

But what is its current status? Genexus Usability has been important from the beginning, and today is still considered important based on the following:

When we start developing, it is not essential to know what the execution platform will be (Hardware, Operating System, Database Managing System, Architecture, and Programming Language to be used).

Developers do not need detailed knowledge of the execution platform. In particular, this aspect helps the recycling of old developers used to obsolete technologies and the addition of new developers with no experience.

Clients get more freedom, because they can change the execution platform at any time and convert their applications with Genexus.

New technologies can be developed in the middle of a large project, and they will be able to use them without any problems.

Genexus automatically integrates the different elements of an application, ensuring its permanent consistency and keeping comprehensive information that is active and always updated.

The systems correction is verified by testing the specifications through complete and timely live prototypes.

This is all very important and most of these features are unique. What is the problem, then? There is no problem, but there is a limitation: Genexus must always be used by professional developers.

Is it feasible in the mid-term to think that all applications should be built by professional developers?

We do not think so: The need to use professional developers only is a result of the limitations in technology!

By lifting those limitations we can consider a much larger contribution from information users, users who know the business - or any other sector - and who do not know (and do not even expect to know) the low level details that are necessary to build a system today.

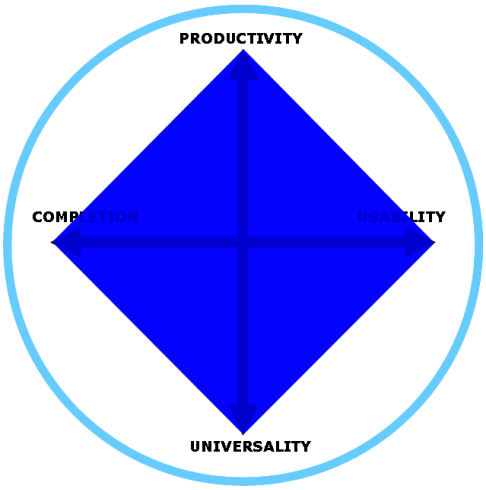
Knowing the problem to be solved will be increasingly more important than the technology necessary to solve it. Systems are becoming more diversified: In the near future, business systems based on bookkeeping, purchases, sales, payroll, stock or ERPs, CRMs, etc. will be a small portion of the applications that users (those hundreds of millions of users) will need. Where is the knowledge to build new applications? We think we can find more of that knowledge in those users than in the professional developers.

For that reason, Genexus will have to continue working to become more "usable": Easier to users that need to solve a problem (a problem they know well) and who are not professional developers.

Specifically, Genexus needs to be friendlier. It needs to reduce the abstraction level necessary to use it, hiding its complexity, and in particular, increasing its declarative component, and adding intuitive and easy to use graphic components, so as to minimize the need to use its procedural component.

This is our big job for the next few years!

THE FUTURE: DRAMATIC INCREASE IN USABILITY



[1] **Breogán Gonda and Juan Nicolás Jodal** graduated from the School of Engineering at Universidad de la República, Uruguay, with degrees in Computer Engineering.
They have conducted extensive consulting, teaching and research work throughout the world.
Their research areas of main interest are Relational Databases, Artificial Intelligence, Automated Treatment of Knowledge and Automated Application Development.
They are recipients of the 1995 National Engineering Award conferred by the Uruguayan National Academy of Engineering.
They are founding partners and, respectively, the President and Vice President of Artech, a company that develops and markets its Genexus product worldwide.

---

[2] **Genexus is a knowledge-based system,** and, specifically, is a system based on a great automated treatment capability of business systems knowledge.

With a solid mathematical foundation, Genexus captures all knowledge contained in users' content and systematizes it in a Knowledge Base.
Genexus knowledge base has a great capability for logic inference: It is able to provide any knowledge stored in it, or logically inferred from those stored in it, at any time. Based on this inference capability, it is able to project, generate and maintain, 100% automatically, the database and programs necessary to satisfy all users' visions known at a given moment.
