AVOIDING THE PITFALLS OF GIS ENGINES CONTAINING A GEODATABASE

_WHITE PAPER

LUCIAD
CONNECT • VISUALIZE • ANALYZE • ACT

Version 1.0

CONTENTS

I. INTRODUCTION

Conventional GIS suppliers often promote their GIS engine claiming that it levers a "powerful geodatabase." The presence of a geodatabase in the GIS engine is thus positioned as an advantage, and even a necessary prerequisite to run a successful geospatial data management architecture.

This may trigger a number of questions when organizations seek a solution for exploiting their geospatial data. For example,

→ Why do GIS engines offer (and even require) the use of an additional geodatabase, when I have already stored my geospatial data in a spatial database of my own?
→ What are the advantages and disadvantages of running a GIS engine that relies on a built-in geodatabase?

The present white paper aims to help you answer such questions

II. WHAT IS A GEODATABASE AND WHY IS IT USED?

In this Section we will focus on properly defining some key concepts.

A. DATABASES

Databases are organized collections of data, modeled in such a way that the data is optimized for specific information queries. Standard databases are able to handle various numeric and character types of data.

B. GEODATABASES

In turn, a spatial database or geodatabase is a database (or extension thereof) optimized to store and query data that represents objects defined in a geometric space. Most spatial databases allow representing simple geometric objects such as points, lines, and polygons. Some spatial databases handle more complex structures such as 3-D objects, topological coverages, linear networks and TINs[1] .

C. WHY ARE GEODATABASES USED?

Historically, the purpose of geodatabases is linked to the very limitations of today's conventional databases. Since many conventional databases are able to handle numeric and character types of data, reference coordinate systems send a traditional database into a spin.

"Data residing in these [conventional] databases is simple, consisting of numbers, names, addresses, product descriptions etc. These DBMSs [Database Management System] are very efficient for the tasks they were designed for. For example, a query like 'List the top ten customers, in terms of sales, in the year 1998' will be very efficiently answered by a DBMS even if the database has to scan through a very large customer database … The database will not scan through all the customers, it will use an index, as you would with a book, to narrow down the search. On the other hand, a relatively simple query such as 'List all the customers who reside within fifty miles of the company headquarters' will confound the database. To process this query, the database will have to transform the company headquarters and customer addresses into a suitable reference system, possibly latitude and longitude, in which distances can be computed and compared … It will not be able to use an index to narrow down the search, because traditional indices are incapable of ordering multidimensional coordinate data. A simple and legitimate business query thus can send a DBMS into a hopeless tailspin" [2]

[1] 1www.wikipedia.org.

[2] S. Shekhar and S. Chawla, Spatial Databases: A Tour, Prentice Hall, 2003

D. GEODATABASES AND GEODATABASES.

The term "geodatabase" can mean different things.

First, it can be used to designate "stand-alone" spatial databases. These can be classical databases such as IBM DB2, Oracle Spatial, MySQL or in-memory databases such as SAP HANA, Oracle TimesTen, and VoltDB to name a few.

Second, it can be used to designate a spatial database within a GIS engine. As a matter of fact, most if not all conventional GIS engines rely on some sort of built-in spatial database. It is this type of built in geodatabase that we will analyze in the following section.

III. GIS ENGINES WITH A GEODATABASE: COMPLEX AND LINEAR DATA TRANSFORMATIONS AND PROCESSING

Conventional GIS engines cannot handle geospatial data with the click of a button. They need to take the geospatial source data through a complex and linear process before being able to process and analyze that data. Let's take a look at a typical enterprise GIS system, where different data sources (vector, raster, real-time, etc.) must be pulled together by the GIS engine.

First, all incoming data that comes from different sources in different formats needs to be transformed into formats that the GIS engine can handle. Specialist tools like a feature manipulation engine (FME) are sometimes used for this transformation.

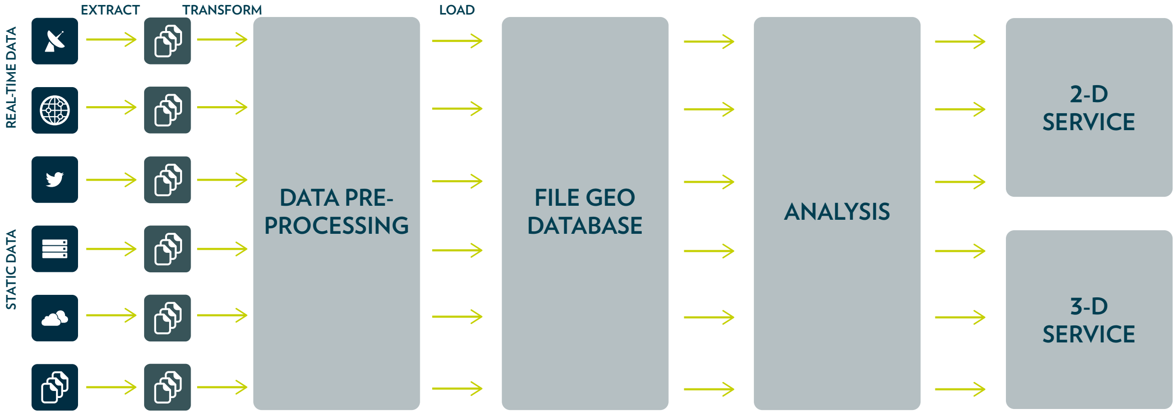


Figure 1: Schema of conventional GIS engine relying on a geodatabase

Second, once all the available data is transformed to formats supported by the GIS engine, the data needs to be "translated" to a single internal format that is often proprietary to the GIS engine vendor.

Third, all the data in this single internal format is stored in an internal geodatabase. It is on this geodatabase that spatial queries are performed.

Fourth, the results of the spatial queries are then exported into a visualization application. Depending on the capabilities of the file geodatabase, some GIS engines will offer 2-D visualization, while others will offer a 3-D visualization.

IV. PITFALLS WHEN USING A GIS WITH A BUILT-IN GEODATABASE

A. INTRODUCTION

GIS engines relying on the schema set forth in Section 3 can bring about a number of serious risks. These risks may remain hidden when the GIS system is used on a small-scale – but will become fully exposed once the system is deployed on a large or enterprise scale. Below we address these risks.

B. LENGTHY DATA TRANSFORMATION CHAIN WITH DIFFERENT TOOLS

As indicated above, all source data must first be delivered to the geospatial engine in a format that it can handle. On numerous occasions we have seen user organizations struggle with legacy software setups where three, four or even more different software products were needed to carry out data transformations. In other words, a whole series of software needs to be aligned for the sole purpose of preparing all the source data in formats that the GIS engine can recognize. This process is not only lengthy and time-consuming – it obviously also inflates the Total Cost of Ownership (TCO) as different systems need to be purchased and maintained. In addition, operators must be trained and well-versed in using the different software products.

C. LOSS OF DATA INTEGRITY / PRECISION

Data transformations are nothing less than a kind of “translation.”

Translations can lead to inaccuracies and imprecision compared to the source. Some conventional GIS vendors even openly warn on their developer web pages that data transformations may lead to up to hundreds of meters of inaccuracy – for a single transformation. In case of subsequent data transformations, the inaccuracy risk increases exponentially.

Linked to the topic of data precision is the topic of data integrity. Sometimes source data is encrypted – for instance to preserve the authenticity or to protect the proprietary nature of such data. Conventional GIS systems need to store data in a decrypted form and therefore cannot guarantee data integrity. An example is maritime charts that often come in the encrypted data format S-63. Many conventional GIS systems do not have the required authorizations from the International Hydrographic Organization (IHO) to handle encrypted S-63 data. Since such GIS systems rely on a geodatabase. This in turn requires converting all data into a uniform, unencrypted format that can be stored in a geodatabase, which ultimately defeats the purpose of the S-63 data encryption.

D. LOSS OF PERFORMANCE

Another logical consequence of the use of internal geodatabases is performance loss. One of the factors that influences the performance of any software system is the number of steps the data must go through to be processed and transformed. The use of internal geodatabases and proprietary data formats mandates many such steps. This lengthy process explains why geo-professionals in large telecom companies, defense organizations, utilities and mining companies spend hours, if not entire nights or days, to load their data into their GIS systems.

The problem of performance loss becomes even more acute when conventional GIS systems are required to process dynamic rapidly changing or real-time data. In that case, by the time the data makes its way to the file geodatabase and is queried, the source data has already changed. At best this leads to severe latencies in the system. At worst – and this happens more often than not – it leads to systems that stall, freeze or crash, because real-time source data is queued and causes a kind of “data overload.” In areas where real-time data is crucial, such as aviation and defense command and control systems, this problem is widely known and recognized.

E. DATA LOCK-IN

With the use of a built-in geodatabase comes a consequence that is proper to any database: the topic of data migration. Conceptually, this can be easily explained. Conventional GIS systems do not bring processing to the data. Rather, they transform and mesh up all data in a database to ultimately bring the data to the processing. And when the processing capability – the GIS engine – changes, lengthy database migrations are a logical corollary. This is commonly known in the geospatial community as the "data lock-in" – the choice for a specific GIS vendor requires the user to store all data in the proprietary geodatabase, leaving the user to continue using the same GIS technology so to avoid a painful data migration.

F. EVERYTHING IS GEOSPATIAL - BUT GEOSPATIAL ISN'T EVERYTHING

GIS engines containing a geodatabase expect an entire IT system to adapt to the GIS engine's logic rather than adapting to the larger IT system architecture. We explain this in further detail below.

In today's world, virtually all data has or potentially will have a geospatial component. The advent of the Internet of Things (IoT) will cause more and more individuals, assets and objects to transmit continuous streams of information – and all of that information can be geo-referenced. At Luciad for instance, we have worked with aircraft engine manufacturers that want to monitor the in-flight engine performance. All engine performance data is streamed in real-time to a ground station, together with the aircraft's position, heading and speed. In that way engineers can overlay the track data of the aircraft with engine performance data alongside extraneous data such as weather information. So there is no doubt: in today's world, "all data is geospatial."

As much as geodatabases allow to unlock the geospatial power of data, they can also inhibit the non-geospatial power of that same data. Consider, for instance, the in-flight engine performance data of the example above. That data may be also interesting to financial analysts looking into the aircraft's fuel burn compared to the payload and the pilot crew's years of experience. For such an analysis, the aircraft's location data is irrelevant. And geodatabases may not be fit for such advanced financial, non-geospatial analysis.

In other words, "all data is geospatial – but the geospatial aspect isn't everything."

Most GIS architectures that impose the use of a geodatabase overlook this critical element. By doing so, entire IT architectures often need to be adapted to fit a GIS architecture, and it would make a lot more sense to use versatile geospatial software components that easily fit into any IT architecture.

G. HITTING THE RELATIONAL WALL

Today, Big Data and IoT are a reality. The IoT results in organizations having to deal with unprecedented streams of data. Processing and storing such data quickly becomes a performance bottleneck when using built-in geodatabases. Perhaps more importantly the analyst who is trying to reveal relations, patterns, or anomalies, will quickly hit the so-called "relational wall."[3] Today's queries performed by the analyst are far more complex than simple range or containment queries. Consider for example an intelligence system keeping track of moving assets. Trying to find out which assets move together, so-called "convoy analysis," quickly results in severe performance issues when using built-in geodatabases. The reason being that these geodatabases are relational databases (RDBMS), which require many so-called slow "join operations" on tables to answer these queries. As such, classical GIS systems with internal relational geodatabases are bad at connecting the dots.

H. SO WHY THEN DO SO MANY GIS ENGINES RELY ON AN INTERNAL GEODATABASE?

As set out below, conventional GIS with internal geodatabases can cause many headaches:

→ Difficulty or even impossibility to ingest custom formats
→ Difficulty (e.g. tracking of large numbers of citizens, cars, or objects) or even impossibility (e.g. radar feeds) to handle real-time data efficiently because of the need for preprocessing in the geodatabase. Adding heavy "event processors" often does not solve the problem and only adds to the complexity
→ Difficulty or even impossibility to deal with today's more complex queries and analysis
→ Lack of extensibility due to the "black-box" nature of the geodatabase
→ Lack of flexibility to adapt to the wider systems architecture

[3] Hitting the relational wall, White Paper, Dr. Andrew E. Wade, Objectivity

So why then do so many GIS engines still rely on a file geodatabase? The answer is simple: legacy. Geodatabases date from a time twenty or thirty years ago when geospatial data was essentially static in nature. At that time GIS was used for cadasters, land management and surveying. Opting for a geodatabase thus was a legitimate and logical choice at the time.

Since then however, many GIS vendors have not adapted their technology to today's dynamic use of geospatial information.

Many if not all of today's classical GIS vendors still work with a geodatabase. They apply a common information model, proprietary for each GIS product. The geodatabase is closed and manages all geospatial objects, rules and relationships. Such "black-box" technology may work for static and individual user applications – but is entirely unfit for enterprise-wide and real-time needs. And yet, that is where everyone is moving: to enterprise-wide sharing and real-time analysis of data.

V. CONNECT, TRANSFORM, VISUALIZE, ANALYZE, SHARE AND ACT – IN-MEMORY COMPUTING WITH LUCIAD

A. NEED FOR A PARADIGM SHIFT

As demonstrated in Section IV, today's world calls for GIS engines to make a paradigm shift away from built-in geodatabases.

As a matter of fact, today's geospatial use cases typically imply intensive use of real-time data. This data includes moving tracks, positions of a person or asset, live imagery and other sources. Thus, the common thread in today's use case of geospatial data is the requirement to connect to, visualize and analyze any data source on-the-fly. To accomplish this, pre-processing and optimizing the data is simply not the answer.

B. LUCIAD: IN-MEMORY ACCESS TO ALL DATA WITHOUT PRE-PROCESSING THROUGH A SINGLE UNIFIED DATA MODEL

Luciad set out to meet the needs of Defense and Aviation users – domains where the reality of real-time data was present much sooner than in other domains. Thus, from the outset, Luciad has purposely chosen an in-memory, native connection to all data. In other words Luciad users do not need to work with a "geodatabase" or "event processor" – synonyms for low performance and instability in real-time operations.

The data flow of the Luciad products is nimble and allows very high performance and fast, precise calculations. That is best illustrated by the fact that all linear processing operations of conventional GIS are carried out in one single, in-memory step with Luciad:

No data transformation step: All data is accessed and processed in its native format, without pre-processing, through a range of advanced data connectors. The specifications of these connectors are open and there is no reliance on any internal proprietary format. Thus anyone (Luciad or its customers) can easily build additional connectors for any type of new data or technology. This is one of the reasons why Luciad was the first geospatial vendor able to offer full commercial support for the AIXM5 standard. Or why it was the first to launch a Geopackage viewer capable of handling the new Geopackage format.

No geodatabase step: The data is not stored in a geodatabase. Rather, an in-memory "unified domain model" does all things at once. All source data is queried at the source, without intermediate storage.

No modeling step for 2-D and 3-D: The "unified domain model" contains a single model for both 2-D and 3-D, ensuring that 2-D and 3-D visualization and analysis is perfectly aligned and coherent. This architecture provides users with a unique feature: switching back and forth between the 2-D and 3-D view seamlessly at any time during the session, and any changes or calculations made in one projection are immediately available in the other projection.

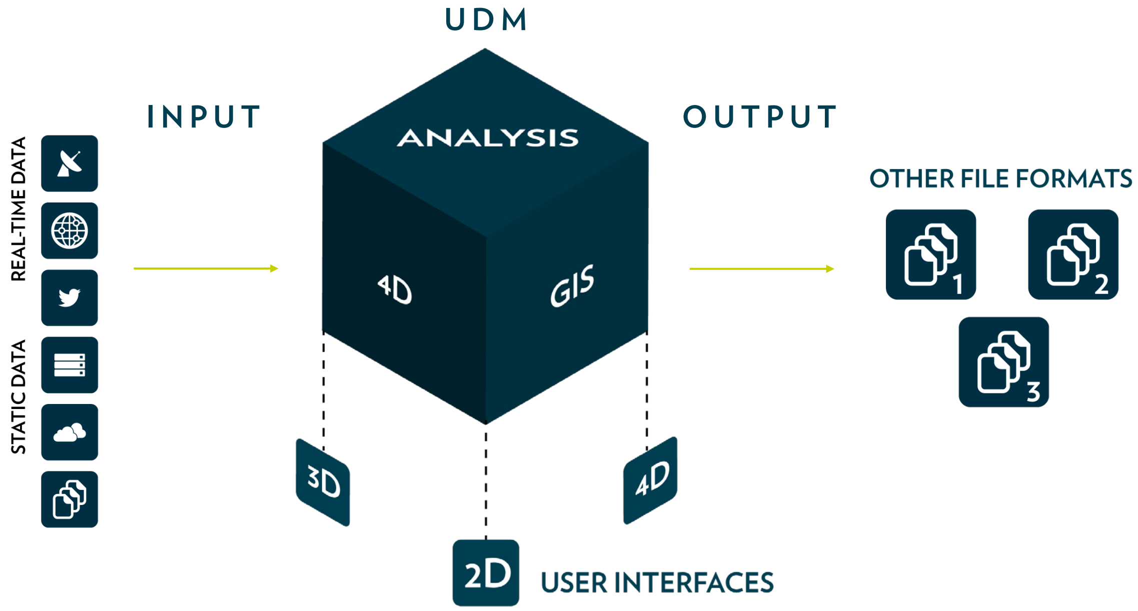


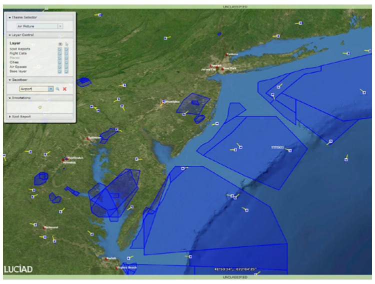
Figure 2 : Schema of in-memory data access by the Luciad products

C. USE DATABASES ONLY WHEN IT MAKES SENSE

Luciad does not advocate against the use of databases. Instead, we are flexible, we do not impose a specific geodatabase. Many of our users rely on databases, and our products come complete with many out-of-the-box database connectors, including connectors for databases such as the ones from Oracle, IBM and SAP. We also do use databases in our products - but only when it makes sense. For example LuciadFusion uses database technology to store and serve pre-processed world-covering elevation, imagery and vector data sets.

What differentiates Luciad from conventional GIS providers is that we are flexible and we do not impose a specific built-in geodatabase. If a customer decides to work with an object-oriented database, instead of a relational one, he is free to do so. Or if using a database does not make sense at all, for example when connecting to real-time video radar or sonar feeds, that is no problem either.

In essence, with Luciad, users can connect to any data source without the need for pre-processing. Users only use a database for the cases that make sense and are free to choose the technology that best suits their project's requirements.



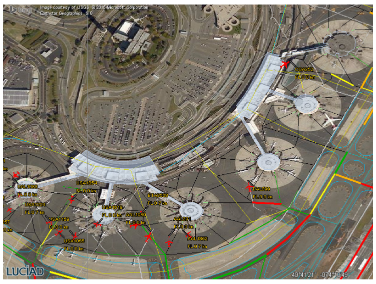


Figure 3: Data Connectors allow users to connect to any type of data, including imagery, base raster maps, vector data such as street data, and domain data such as 3-D buildings, maritime maps of coastal area, airport infrastructures, etc. – in 2-D and 3-D.

USEFUL REFERENCES

Please contact Luciad for a list of references from customers who have successfully replaced their database system with a Luciad solution.

MORE INFORMATION

For more technical information or to find out more about our other products and services, please contact us at **info@luciad.com**.

www.luciad.com **U.S. & Canada** +1 202 507 5895 • **Europe & Rest of World** +32 16 23 95 91