DATABASE CONVERSION - ONE EXPERIENCE

Matthew Burnham
Michael Fortin
Daniel Jaquish
Hughes Training, Inc.
Arlington, TX

ABSTRACT - A number of factors have come together in recent years to make the reuse of visual databases a sought after goal. Primary among these has been the rising cost of database development due to ever larger real-world gaming areas and ever increasing system capacities. On the more positive side, the basic technology of the available image generation platforms have begun to share some common architectural features which make conversions from one format to another a somewhat more manageable task than in years past.

This paper will discuss the implementation issues, and successful completion, of one specific conversion effort. The requirement is in support of an advanced Weapons Tactics Trainer which utilizes a Lockheed Martin CompuScene VI for the out-the-window images and a Silicon Graphics RealityEngine2 for infrared imagery. Correlation between the visual and IR images is required to be as close as possible, preferably exact.

The points covered will include the development and implementation of the conversion algorithms and discussion of the numerous issues which result from supporting two platforms from different manufacturers. These include differences in texture map formats and utilization, polygon capacity, management of geo-centric databases, material code correlation and Z-buffer priority solutions. Also discussed will be the lessons learned that will be applied to future conversion efforts, including the use of SIF data for visual and sensor applications, and the practicality of developing true plug and play databases that will be compatible with a wide range of image generation platforms.

AUTHORS:

Matthew Burnham started as a Database Engineer with Evans & Sutherland in 1987. While there he worked on various exploitation and marketing efforts including database conversions for several different image generators and IG optimization efforts. He also worked for several years on the DIS demos for the I/ITSEC conferences. He has been at Hughes Training Inc. for two years leading the database tools effort. The main emphasis of this tools effort is platform independent database production and conversions, allowing databases to be converted between and used on various image generation platforms.

Michael Fortin joined Hughes Training, Inc. (then Rediffusion Simulation) in 1974. In that time he has been involved with a number of technical areas in visual simulation, primarily related to database issues. These have included database production, exploitation and tools development. He has also held positions in Technical Marketing and Product Management. He is currently in the Advanced Programs Systems Engineering group pursuing new business opportunities. Mr. Fortin received his bachelor's degree in math from Florida State University and has served as a U.S. Naval Aviator.

Daniel Jaquish has been a Database Engineer since 1981. He has designed databases for ten generations of image generators and has helped drive the tool development necessary to keep pace with this rapidly expanding technology. Much of his work has involved large-scale terrain generation and development of texture and material effects to enhance the realism of software-generated terrain. He spent three years with Rediffusion Simulation Ltd., of the United Kingdom testing and refining graphical modeling tools for general database construction and developing concepts for material coding and classification which are fundamental to current database conversion strategies.

DATABASE CONVERSION - ONE EXPERIENCE

Matthew Burnham
Michael Fortin
Daniel Jaquish
Hughes Training, Inc.
Arlington, TX

INTRODUCTION

In mid-1995 Hughes Training (HTI) determined a need to use two different image generation devices in the same advanced jet Weapons System Trainer (WST). This configuration was selected in order to reuse software previously developed for the LANTIRN and Maverick avionics on the Air Force Unit Training Device (UTD) program.

The out-the-window (OTW) imagery would be supplied by a three channel Lockheed Martin CompuScene VI (C6), the LANTIRN and Maverick infrared imagery for the multi-function displays (MFD) and HUD by a Silicon Graphics Inc. (SGI) RealityEngine2 (RE2). The SGI also served as the host computer for the simulator (Figure 1).

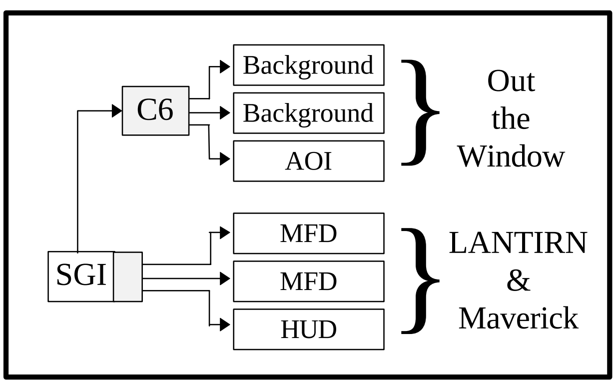


Figure 1 - WST Block Diagram

In order for this configuration to provide proper training to the flight crews it would be necessary for the same environment database to be operating in both systems with a very high degree of correlation. The environment database specified for the WST included approximately 240 geo-cells (one degree of latitude x one degree of longitude) and several high-detail airfield and target models.

There were several possibilities for creating the two databases:

- Build two separate databases
- Build an SGI database and convert it to C6
- Build a C6 database and convert it to SGI
- Build a common database and convert it to both C6 and SGI

It was determined that a conversion of the C6 database to run on the SGI was the best approach due to the advanced features inherent in the image generator. The goals for the conversion were to achieve a high degree of correlation, 100% if possible, and to simplify the task as much as was feasible through automation.

PREVIOUS EXPERIENCE

Previous conversion efforts had primarily involved the moving of models, simple and complex, from one platform to another. Typical applications were Feature Model Library (FML) components for populating DFAD derived databases, complex moving models of aircraft and vehicles used in conjunction with threat databases, and large airfield models which had been hand built, often at great expense. This experience included specific converters or translators for E&S models to SGI formats, E&S models to C6 formats and SGI models to run on the C6. Conversion efforts of this type have been the most common to date. The intent is to benefit from the value added portions of the origin database. Due to the efficiency of current terrain tools, it is the costly hand built (i.e. man-power intensive) portions of a database which are the most beneficial to convert.

As more of these conversion utilities were developed a number of trends became evident. While it was possible to reuse some of the software routines from one utility to the next, it was also seen that there was room for a much higher degree of commonality and generalization to take advantage of similar implementation features. The situation that was developing is simply depicted in Figure 2. There was a

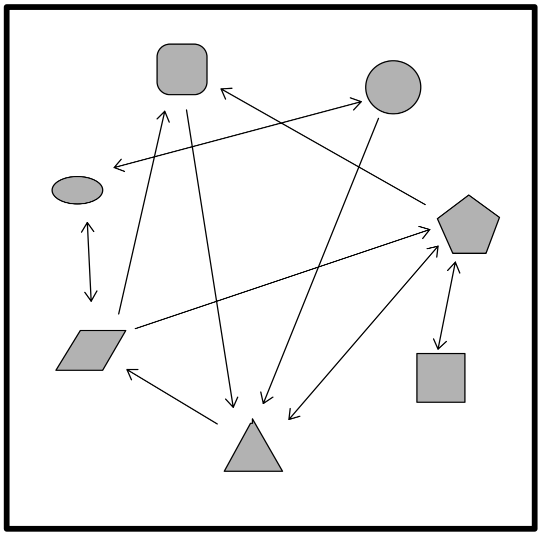


Figure 2 - Multiple Conversions Between Platforms

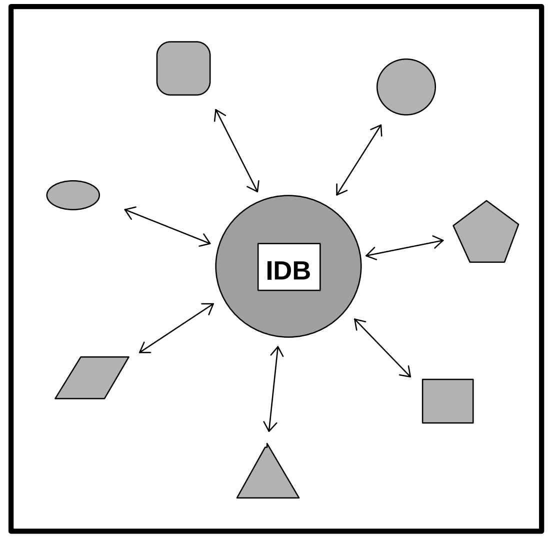


Figure 3 - Internal Database Concept

potential for *n* x *(n-1)* utility programs where *n* is the number of origin and target platforms. Techniques to simplify and combine these utilities lead to the concept of the Internal Database (IDB).

Internal Database Concept

The purpose of the IDB would be to capture the common aspects of the conversion utilities that had been developed and incorporate them into a more general conversion tool set. A major feature of the IDB had to be extensibility so that future platforms and applications could be added efficiently without major redesign. Figure 3 indicates the difference between the individual utility approach shown in Figure 2 and the efficiency inherent in the IDB. The number of converters required drops from *n* x *(n-1)* to *2n*. There are also more possibilities for reuse on the individual converters, so the total effort is further reduced.

Part of the design process for the IDB included looking at other attempts at large scale database conversions and to examine related efforts in object oriented database manipulation. Project 2851, and the resulting Standard Interchange Format (SIF), was one area that required study. One possibility could have been to use SIF as the core of the IDB, but there are a number of basic differences between the two approaches.

The most obvious is that SIF is an interchange media intended to store data in a standard format while it is awaiting transit from one system to another. Translators are used at both ends of the interchange to take the database to and from the native format. As shown in Figure 4, the IDB was conceived to be an internal (to the computer) functionality. Databases are not written or stored in an IDB “format”. The IDB is more like an application programmer interface (API) that provides connectivity between native formats. This approach is intended to minimize the loss of fidelity which results from the translation steps by reducing the number of these steps. Functionality can also be included which will make the translations more robust. In general, it was seen that the more functionality that could be included in the IDB, the less important the individual formats became. Examples of similar approaches are seen in the HTML and Java API’s now coming to the Internet.

Another difference between the IDB and SIF is the candidate platforms they are intended for. Project 2851 was faced with the task of providing something for everyone. The plan was to support a very wide range of systems, with an even wider range of features and capabilities, so that all platforms could participate. This made for a very large first bite.

Conversely, the IDB will initially support a very narrow group of systems. Specifically, those systems being used on current simulation programs for which we are providing databases. As mentioned above, extensibility to additional systems is a critical part of the design, but the actual implementation will take place in small, manageable steps.

As part of the IDB design and implementation it was necessary to look at the factors which drive a

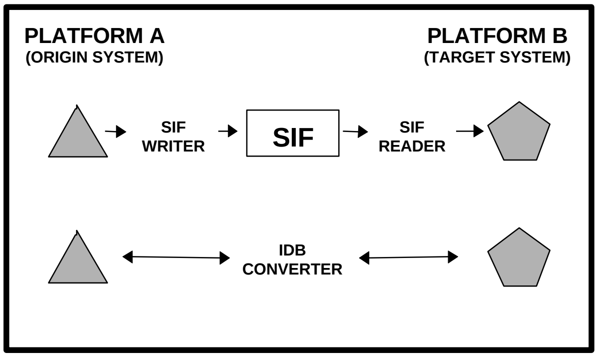


Figure 4 - SIF vs. IDB Conversion

conversion effort to ensure that all of the issues were accounted for.

DRIVING FACTORS

In considering the database conversion effort as a production process a series of questions and decisions must be resolved. Depending on the success of the conversion effort, it may be necessary to go through this process more than once. We will look at these issues in a theoretical sense and then examine how they applied to the C6 to SGI conversion in particular.

Theory

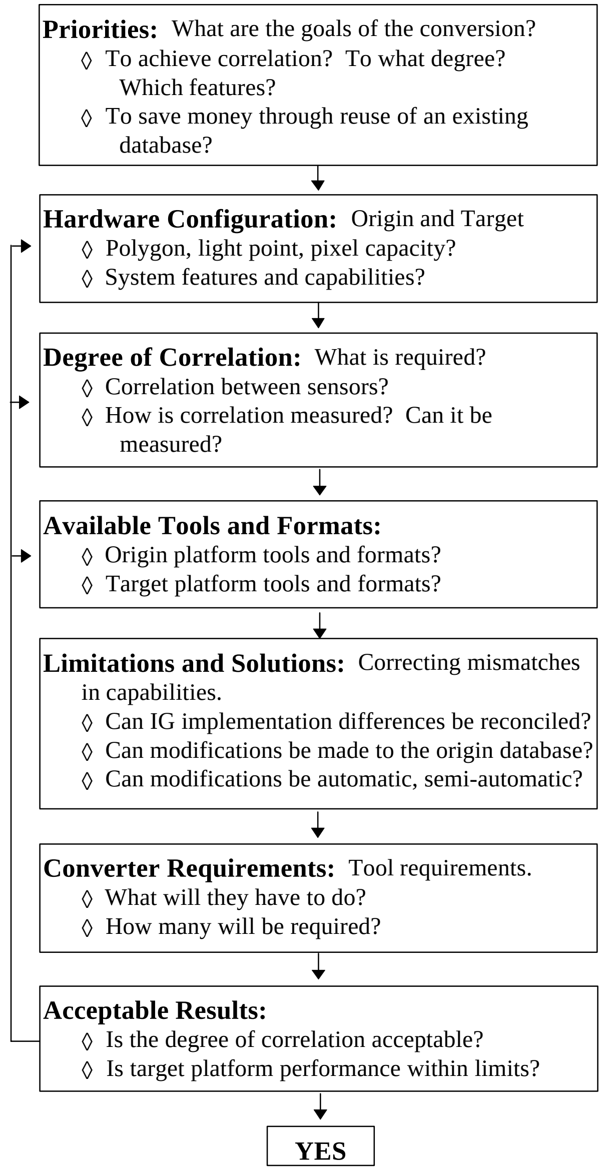
The decision tree for a typical conversion effort is shown in Figure 5. Most of these questions will have to be answered for any type of database conversion.

Experience

In applying this methodology to the CompuScene VI to Silicon Graphics conversion a number of problems were uncovered which were seen as typical of such a complex task.

Priorities - The reason for the conversion was to reuse the software that simulated the functionality of the infrared sensor suite in the aircraft. The implementation of the various tracking functions being the most critical aspect of the effort. This had been done successfully on the UTD program which uses the Silicon Graphics as both the visual/IR system and the simulator computer.

While database cost is always a factor in any visual simulation, the primary goal of the C6 to SGI conversion was to achieve the best possible correlation between the visual OTW imagery and the IR sensor imagery.



Hardware Configuration - The out the window visual system is a three channel CompuScene VI with a dome display system. Two channels provide the background image and the third channel supplies the imagery for a head-tracked area of interest (AOI). The system is configured to provide approximately 2,000 polygons per channel at 60 hz.

The Silicon Graphics system selected for the infrared sensor simulation is a RealityEngine2 providing three channels of imagery at 30 hz. The in-house developed RightView real-time software provides the simulation applications. The channels are typically magnified with an effective field of view of 3°. In configuring the system it was determined that the SGI could emulate the C6 imagery within these performance parameters.

Degree of Correlation - There are a number of correlation issues between visual and IR that occur even if the same IG hardware is used for both. The most obvious is the relationship between the visual color and the emissivity-related gray shade. In both systems the IR intensity is calculated in real-time from material codes in the polygons and objects. It was therefore necessary to ensure that the material codes were included in the conversion.

Another issue is the difference in magnification. Where the visual is 1:1, the IR sensors can range from 1:1 to 10:1 depending on the system. This requires that high-detail models and terrain be visible at very long ranges.

When the pilot looks at a tank 10 miles away with his IR sensor he expects to see the high-detail version (50 - 100 polygons) rather than the low-detail (5 - 20 polygons) version which is displayed in the visual image at that range. Additionally, the terrain clutter around the tank must also be in high detail so that the target acquisition task is realistic. This creates a correlation disparity between the visual and IR regardless of the image source.

Available Tools and Formats - HTI is a licensed user of the Lockheed Martin TARGET tool set for building CompuScene databases. While the final database formats are considered proprietary there are a number of interim formats in the TARGET database generation process that are available. These included terrain files, model files and image (i.e. texture) files and their supporting configuration files. The basic components of the database are included in these files, but various merging activities take place in the later stages of the production process and/or during real-time processing by the image generator. These functions had to be emulated as part of the conversion.

Access to the file formats on the SGI side of the conversion was much less of a problem. The real-time software for the sensor simulation would be RightView which had been developed in-house. This did not mean that formats and functionalities could or should be readily changed, however. The file formats for the database are designed to interact with the real-time functions of the system. They are also standardized across several systems and applications. It was desirable to maintain those product standards and not create a special version of RightView or the database for this particular program. Even working within those limitations, however, this situation provided a great deal more flexibility than could be expected with systems from an outside vendor.

Limitations and Solutions - Figure 6 is a representation of the limitations, in the form of capability disconnects, that can exist between platform features. The major limitations encountered and the chosen solutions included the following:

- Level of Detail (LOD) **-** The C6 uses a very effective continuous level of detail mechanism for the terrain portion of the database. This function brings in the next higher level of detail polygons in the same plane as the lower detail *parent* polygon. The vertices of the new polygons are then moved over time to their high LOD position. This is a very smooth effect with no popping or gaps visible in the image.

  The SGI/RightView system had not yet incorporated a continuous level of detail function so an alternate implementation had to be found. Considering the magnification problem relative to level of detail and the polygon performance of the SGI it was decided to have only one level of detail for the terrain. It was determined that the system could handle the polygon load at the field of view and update rates selected.

- Tile Size **-** Square terrain areas called "tiles" are brought into on-line memory by the database management mechanism. The size of these tiles,

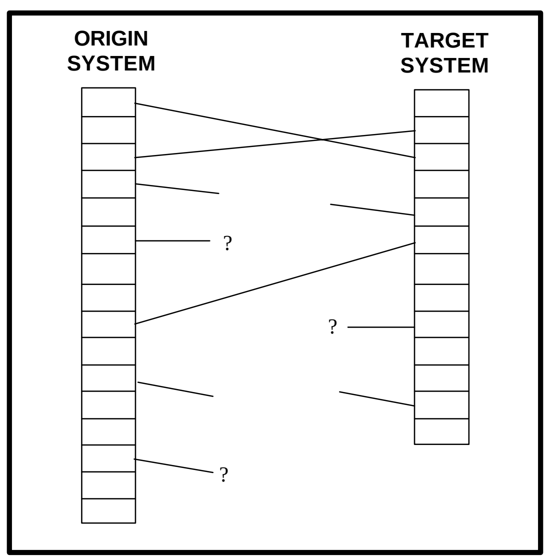


Figure 6- Mismatched System Features

and the number of associated polygons, is a major component in the design of the database and its ability to function efficiently in the image generator. In this case the tile size needed to optimize the C6 performance did not match that required for the SGI. It was therefore necessary to re-tile the terrain and the associated culture features. This involved creating a clipper function that split the C6 polygons where they crossed the new tile boundaries.

- Real-time Features - The C6 performs a number of database related functions in real-time. Among these are procedural universal features (PUF's) which are randomly placed objects to add scene density. This function was not yet implemented in the SGI/RightView system. Very few of these features were designed into the C6 database. It was decided to remove the PUF's that were there and replace them with universal features (UF's) which are similar except that they are positioned off-line in the TARGET tools. It was then possible to emulate this function in the conversion so that the features would be included in the SGI/RightView database.

- Features Added in Production - The early stages of the TARGET production process for C6 databases uses a number of pointers to features which are created or placed in later stages. These included the placement of areal and lineal features such as forest areas and roads. The available C6 formats include a 2D definition and placement of the feature, but the polygons are mapped to the terrain skin in a later process. This function was duplicated as part of the conversion of the terrain files. A similar function in the TARGET tool set takes areas defined as forest canopy and raises them a predetermined distance to create a 3D forest effect. This function was also duplicated in the conversion.

- Hierarchy - Not surprisingly, the C6 database structure, which determines how database management and level of detail are handled, did not apply directly to the SGI/RightView needs. This was particularly true after the re-tiling operation mentioned above. It was necessary to create a new database hierarchy on the SGI side which provided optimum performance efficiency. Related problems included the way the two systems calculate and

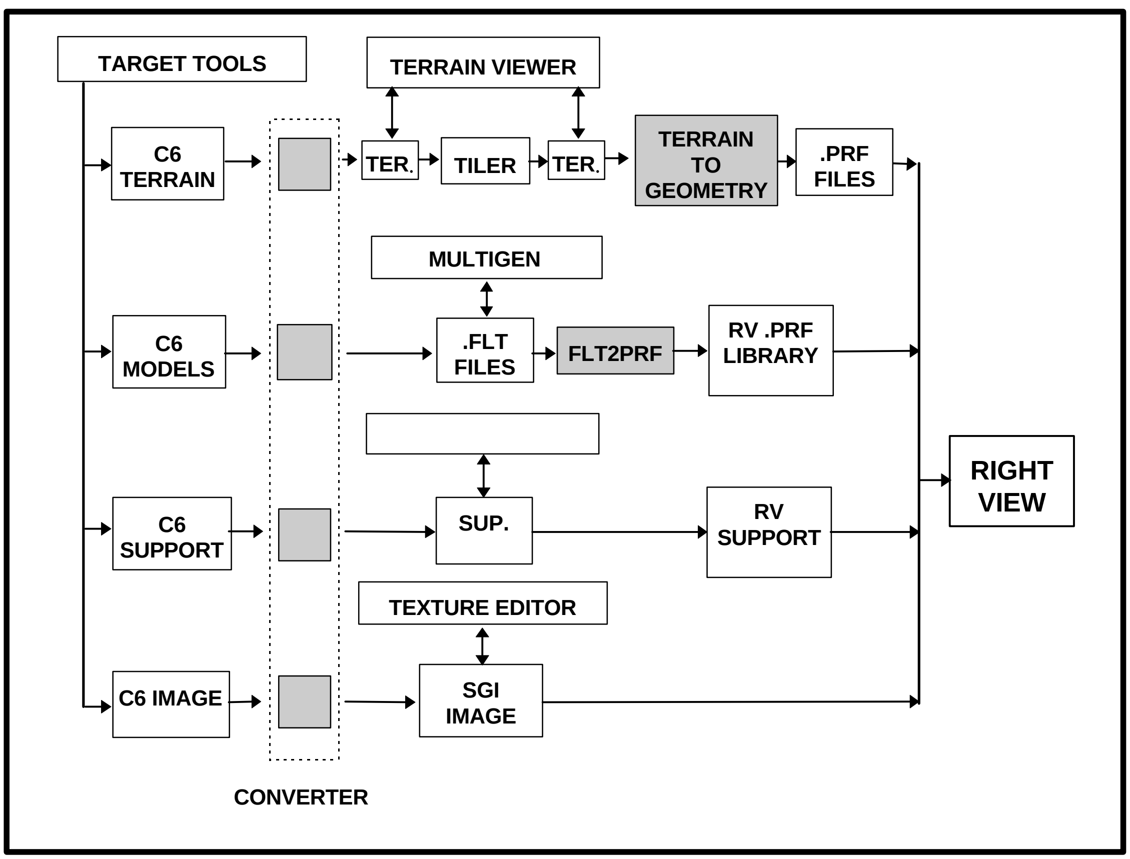


Figure 7- Database Production/Conversion Process

implement level of detail changes for models.

Converter Requirements - There are four major areas of converter functionality. These are terrain, models, imagery and support files which manipulated the other three. The relationship of these functions is shown in Figure 7.

- Terrain Files - The terrain to geometry process shown is an in-house developed tool for generating terrain from more abstract data, including DMA DTED and DFAD. The tool normally works with raw DTED elevation grid data, a filtered version of the DTED data and a polygon skin derived from the filtered data. Cultural features are positioned on the terrain skin per DFAD references. In the C6 case the high level of detail terrain skin is used directly without the two earlier steps.

- Model Files - C6 model files are converted to .FLT format where they can be viewed, and edited if necessary, with the MultiGen 3D editor. The models are then converted to .PRF, the native RightView database format. This is the normal technique for moving models between systems.

- Support Files - The C6 makes use of various support files which define how characteristics are applied in the database. These include color tables, texture placement files and material files. The applicable data is associated with the polygons during the final compiling of the database. Any editing of these files is typically done in ASCII.

- Image Files - The conversion of the texture maps require a conversion from the C6 .PIC file into a .RGB format. It is then necessary to remap the C6 transparency function into the alpha portion of a .RGBA or .INTA map. Texture editing, if required, is done with MultiGen, Adobe Photo Shop or an in-house developed texture tool.

Acceptable Results? - Yes! To date there have been no serious problems encountered with either the performance of the converted database or the degree of correlation achieved. Small problems have arisen that were not anticipated in the original approach. In particular, differences in the Z-buffer priority implementation caused problems which sometimes had to be accounted for. This was due to a need to offset coplanar polygons in the SGI database in order to avoid Z-buffer resolution conflicts. In some cases the offset polygons created a correlation conflict. The

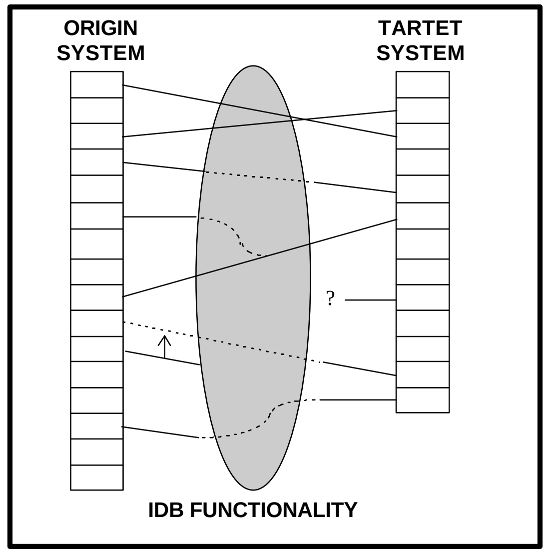


Figure 8 - IDB Connections Between Systems

solution was either to inset the polygons into the underlying polygon or to recess the underlying polygons so that the overlay correlated with the origin polygons. The solution chosen depends on the application.

As mentioned above, there are differences in the level of detail switching mechanisms between the two systems. While this was expected, it was sometimes a more visually apparent artifact than had been anticipated and had to be accounted for by adjusting the transition ranges.

LESSONS LEARNED

Several valuable lessons were learned and areas for additional study were identified during this exercise. The most important of these is the viability of the Internal Database Concept, not only as applied to conversions, but to the overall database production process. The geometric shapes in Figure 3 can represent source data formats (e.g. DTED, DFAD, SIF) as well as tool formats (e.g. .FLT) as well as image generator platforms.

The IDB also provided the connectivity at several levels which allowed the C6 to SGI conversion to be very nearly automatic. Little work, other than validation, is performed on the SGI/RightView side of the process. Figure 8 represents how some features on the origin side of Figure 6 were modified slightly to enhance the conversion while others were remapped into similar, but not necessarily identical, features on the target side.

Other lessons include:

- Absolute correlation between dissimilar image generator platforms is probably impossible at the detail level. This begs the question of levels of correlation short of 100%. Concepts like Engagement Correlation, Operational Correlation, Maneuver Correlation, etc. might quantify degrees of equivalency that could be used for many training exercises, but still fall short of an absolute, 100% correlation of all features.

- It is important to consider changes to the origin database. This greatly extends the possible solutions to limitations and disconnects that are encountered.

- Conversion from an existing origin database has a great deal in common with creating a new database from DMA data. Many of the same issues apply in the design of the target database. It also means that the conversion effort will utilize many of the conventional database generation processes once the converters are in place.

- The automatic functionality of conversion tools like the IDB may lend themselves to adding or improving features when the target system is a more capable and/or later technology platform than the origin system (Figure 9). Higher resolution texture maps could be applied or higher fidelity roads laid on the terrain, for example.

- There are a number of business and programmatic issues which must be resolved. Conventional database review and acceptance practices will not work as well when there are effectively two organizations responsible for the database, the one that created it and the one that converted it. Tracking the cause of a problem, like a missing building or terrain skin irregularity, will require some effort. Similarly, it is very difficult to estimate what a conversion effort will cost without looking at the origin database in some detail, even after the converters are in place.

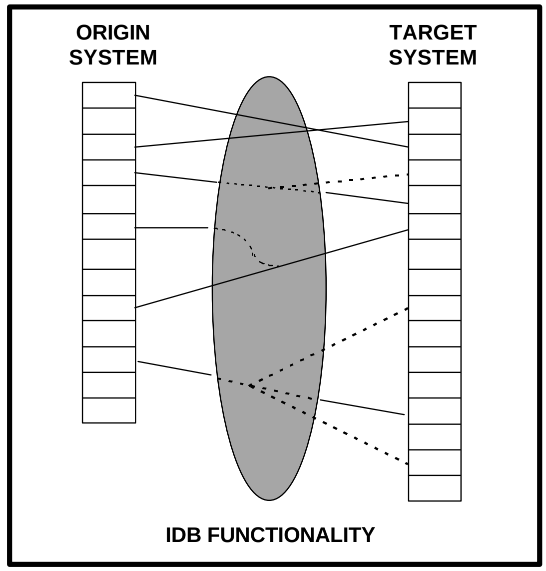


Figure 9 - IDB Enhancements During Conversion

At the time of this writing additional conversion opportunities are being pursued, including one that will be even more demanding than the one described here. Despite the complexity, however, it is anticipated that the overall effort will be considerably reduced this time due to the extensibility of the Internal Database.