

Generic Organisation Data Integration Solution: the fast and convenient way to integrate data.

Richard d'Anjou & François Martel

CGI Group Inc.

5400, Boul des Galeries

Bureau 400

Québec (Québec)

G2K 2B4

(418) 623-0101

richard.danjou@cgi.ca, francois.martel@cgi.ca

Abstract

The best solution to making tactical and strategic decisions at a coalition level would be to completely integrate information systems in a seamless manner without considering the various participating countries. Various technological solutions offer the required functionality to accomplish this.

Unfortunately, even though considerable efforts are currently being deployed to make this possible, presently it is impossible to rapidly integrate information systems that would allow the existence of C2 applications to manage coalitions formed by any NATO member country.

Given the requirements of countries to rapidly intervene in theatres of operation, often jointly with many countries, many army corps, it would be important to have a transitory solution that would allow the rapid integration of the various C2 applications that are used by the various involved countries, until all countries can standardise their respective applications. The Operational Data Store (ODS), is the data structure that integrates heterogeneous data coming from various sources into a coherent set of data and serve as a data source for C2 applications.

In this paper, we will present a solution that uses an ODS to rapidly integrate C2 applications from various countries in order to provide C2 functions to a coalition.

1. Background

The ideas developed within the Generic Organisation Data Integration Solution (GODIS) were inspired from the ideas of the Army Integrated Management Environment (AIME). The AIME project is a multi-phase initiative aimed at implementing a generic technological platform that will favour the integration of data emanating from administrative and operational systems of all backgrounds into a common ODS from where they can be analysed with specialised tools.

Following the development of an organisation data integration solution for the Department of National Defence, an initiative was undertaken to make the solution more generic. The project aimed to modify the existing components in order to make them independent of the military context and of the underlying technology. Performance improvements were also considered with the intent of making the product more reusable.

The major goal of that development was to provide a generic framework of the solution to allow organisations to implement an ODS without having to code everything from scratch. The resulting product (GEDIS), is a reference implementation to prove the concept that can directly be extended to organisations that choose this way of managing data.

2. Data Integration VS Application Integration

The market offers a wide range of products that allow enterprise application integration. CORBA, SOAP, .Net and XIS are but a few examples of this type of infrastructure. However, this integration approach requires considerable standardization efforts, but do provide good results at the application collaboration level. Furthermore, this approach introduces new problems relating to data ownership, validity and refreshing.

For these reasons, it is often difficult to use existing systems in order to integrate applications. In fact, legacy systems are often designed to work in stand-alone mode, as stovepipes. The passage to collaborative/network mode is often complicated, sometimes impossible.

Data integration is a much more simplistic approach with respect to bring about inter-organisation collaboration. The approach is based on the principle of building a data structure that duplicates the structures of the operational systems of the various organisations and thus linking them together in a coherent manner. This new structure has no ownership on the data and has a relative control of their validity. However, data integration is a much more rapid process than application integration since it does not require legacy system standardization. Hence, this structure offers the ability to provide information with a global vision emanating from multiple source applications, all within a relatively short timeframe.

In the context of military operations involving many countries, it is currently difficult to have such a global vision through application integration given that the standardization process has not yet been completed. For this reason, countries wishing to collaborate rapidly had to set-up an infrastructure allowing them to rapidly integrate data.

The solution we will present to you is the result of strenuous work aimed at defining and designing a very rapid means of integrating data from multiple, disparate sources.

3. ODS Concept

The ODS concept emanates from Bill Inmon. This concept is based on an even larger infrastructure known as the Corporate Information Factory (CIF) that is a management framework for enterprise data [Inmon, 1998]. The ODS is a data structure that groups together data from an organisation's operational systems and that is defined as:

- **Subject Oriented:** an ODS is designed and organised around the major subjects of the corporation. The major subjects of a corporation are typically such things as CUSTOMER, PRODUCT, ACTIVITY, POLICY, CLAIM and SHIPMENT. The ODS is not organised around any specific application or function.
- **Integrated:** The data found in the ODS is an aggregation of detailed data found in the legacy systems that feed it. As the data is pulled into the ODS from the legacy systems, the data are fundamentally transformed into a consistent, unified whole. The transformation and integration of detailed legacy data results in a truly integrated, corporate-wide understanding of data as it resides in the ODS.
- **Volatile:** data in the ODS is updated on a regular basis. Every time the data in the foundation source system – the legacy system – changes, the ODS needs to be updated.
- **Current Valued:** data in the ODS is quite up-to-date; there is very little, if any, archival data found in it. If, for whatever application need, archival data is found in the ODS, it is never more than a few days old.
- **Detailed:** data in the ODS serves the operational community and as such is kept at a detailed level. The detailed level is for a given user community. The ODS must not summarise data for the user community of the ODS.

4. Store and Forward Concept

Store and forward is a means of transforming information coming from source systems and transferring it to a target system [Inmon, 1995]. At first, data are taken from source systems (enterprise legacy operational information) and put into a staging database. Once staged, data are then integrated and stored within the ODS database.

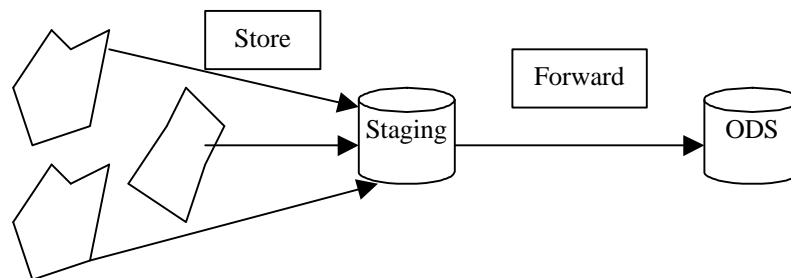


Figure 1 Store and Forward

During the store operation, all of the information is extracted from the source systems and copied to a first database called Staging. This database stores the source system information in their entirety. In this intermediate database, all of the data can be accessed independently from their original sources. This allows a standardized manipulation of data during the transformation operation.

In the second operation, data are integrated, unified and then sent to another database, the ODS.

The result of the two successive operations will give an ODS where data validity corresponds to the same validity of data at the time of its extraction.

5. Store and Forward with Delta

The conventional store and forward technique provides for feeding an ODS, but its subsequent maintenance is very difficult. In fact, this way of proceeding takes an image of the source systems in order to create an ODS and another image must be taken in order to have an up-to-date ODS.

In order to facilitate the ODS update process, a delta section is added to the model. The delta is used to establish the difference between the actual source system data and the latest version of data stored in the ODS. In this manner, information that has been loaded and that is still valid will not go through the integration and transformation process.

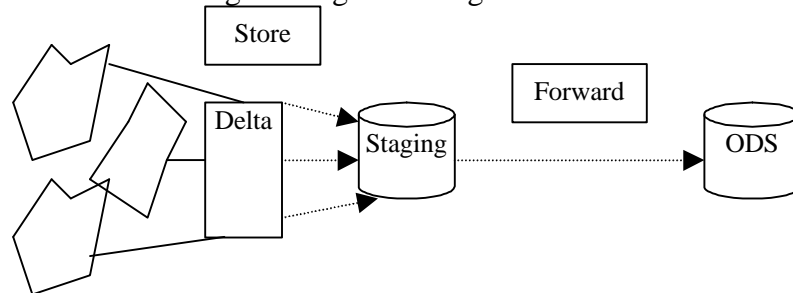


Figure 2 Store and Forward with Delta

The data delta considerably reduces the data load going through the system. Hence, it is possible to quickly, almost real-time, refresh the data. These data, once refreshed in the staging database, are sent alone to the ODS during the forward operation.

In this context, the Store and Forward operations need not be executed sequentially. In fact, we could imagine that the store operation takes place many times while the forward operation occurs only once. For example, the store process could operate all day to provide a continually up-to-date staging database and the forward operation could be executed at night.

6. Data flow of Data Integration in a Coalition C2 Context

In the context of coalition C2 applications, data will flow in a single direction, that is, from source systems toward the target system. It is then impossible that the data change in the ODS have an impact on the source data. In the context of an ODS, operational system data cannot be modified by other operation systems since it is only the original system that can ensure its data's integrity. Furthermore, C2 systems linked to the ODS are conceived for tactical and strategic decision support and should not be adapted for

updating information. The results of this first flow of information is the creation of the ODS, this will then provide a global view within a coalition C2 tool.

The main advantage of this way of doing things is that countries can go about using their C2 applications normally. The solution does not yet, however, support the backward flow of information or the direct flow of information between countries

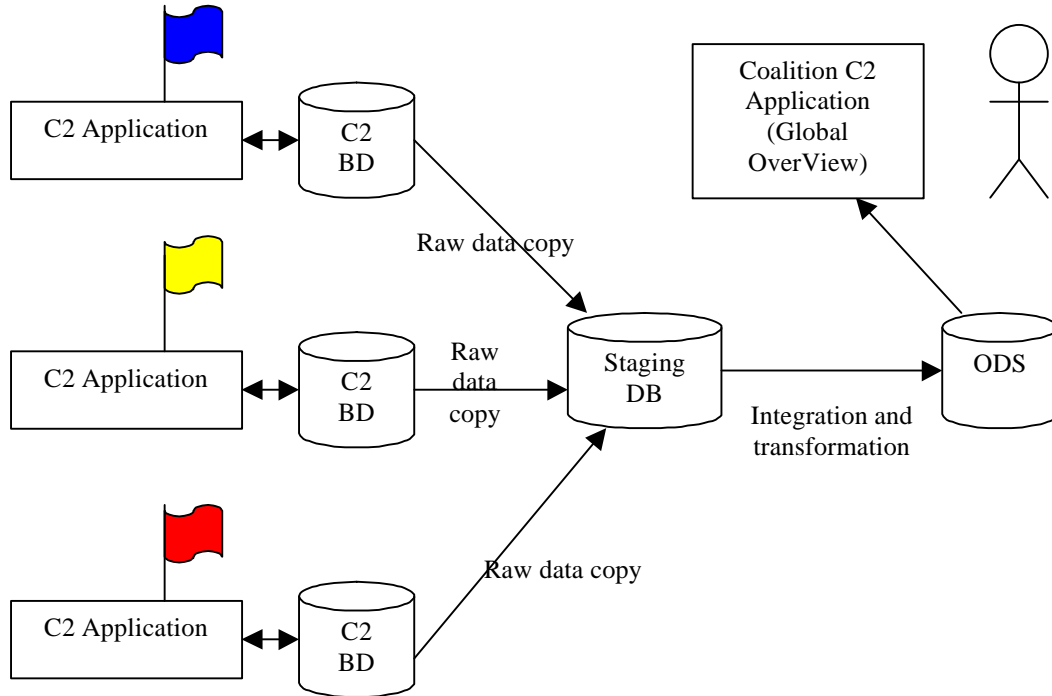


Figure 3 Data Flow from Source to ODS

Figure 4 shows an example of using data from many systems in order to obtain a common view of the situation. In this case, locations.

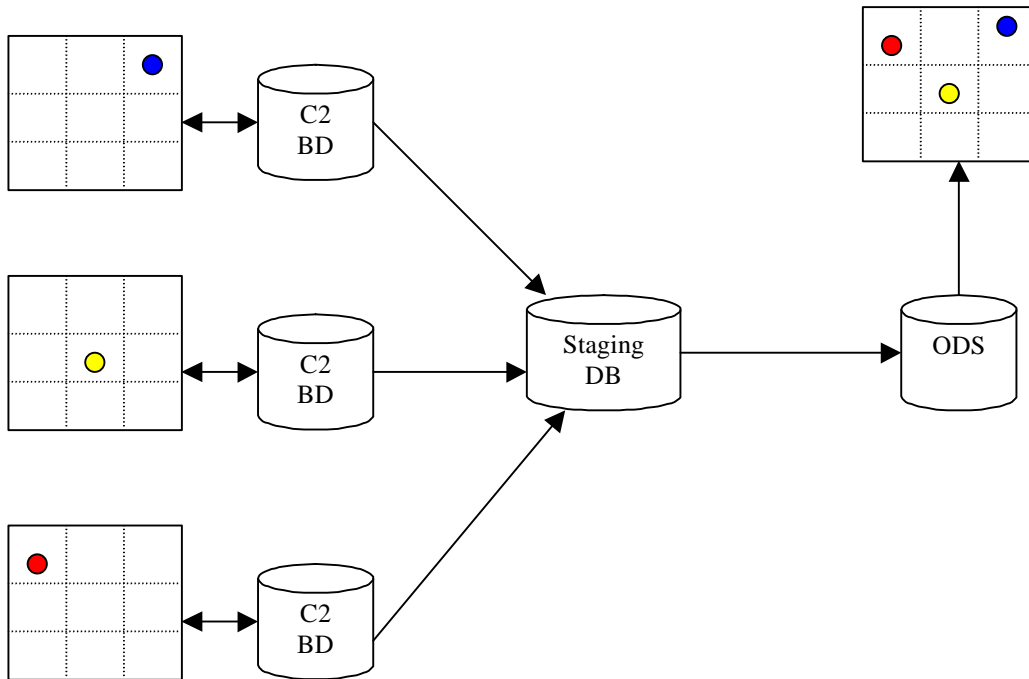


Figure 4 Example : Location

In order to support the return of information, it is possible to add a C2 mart to the solution. That is, a C2 data repository that can be used by all. In this context, the integrated data becomes an entity that is distinct from the raw data: the source systems do not retain ownership of the data and cannot modify it, they can only consult it. The C2 mart can be a separate database, or in fact be the ODS or yet again a service of the global C2 application. Furthermore, the global C2 application could be enhanced with new data or analysis results from the C2 mart. In this way of doing, it is not necessary that the source C2 applications integrate the data mart data. It would be possible to use the data as is without displaying it within a source system.

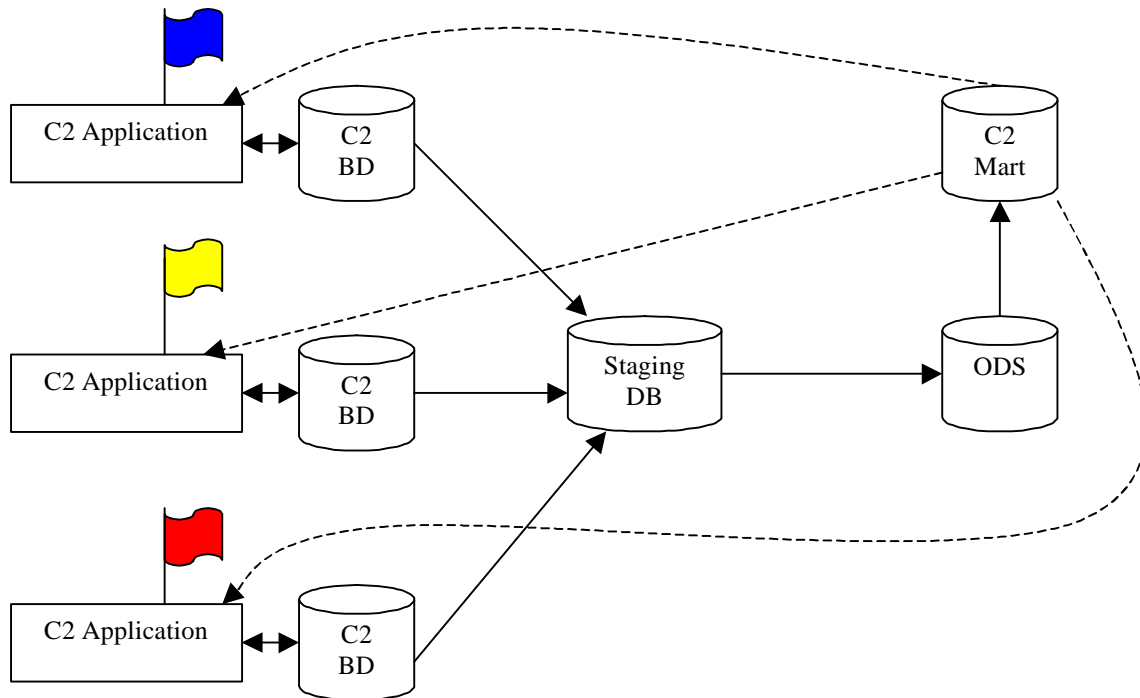


Figure 5 C2 Mart

In the event that a country would have responsibility for Command and Control, the ODS allows using integrated data as if it were operational data. Hence, the C2 application of one country could be reused in order to have a global view.

7. GODIS

To produce an ODS that would correspond to that which was defined previously, we created a number of components within an architecture pattern. This architecture pattern supports the implementation of a 'store and forward with delta' concept for the construction of an ODS. Obviously, this pattern is not exclusive to C2 applications, it may also be used by any organisation wishing to rapidly create an ODS.

Structure :

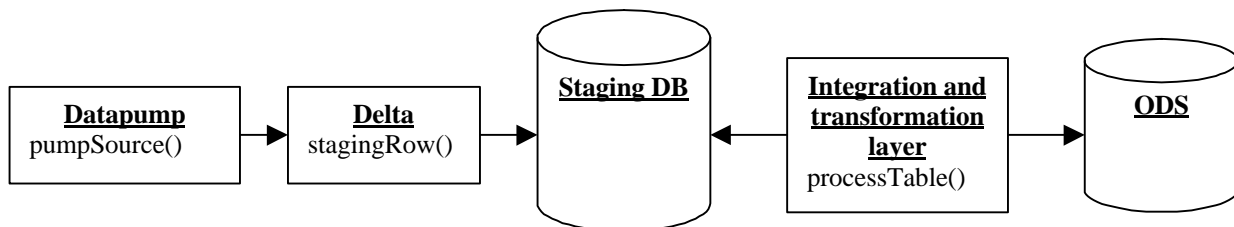


Figure 6 GODIS Structure

Participants :

Each of the following components is run as a Thread;

- Data pump: Retrieves the content of a source system (DB)
- Delta: Stages each of the rows of the source system DB and determines the delta for existing data
- Staging DB: Contains all the information in a single format, accomplishes synchronization between the Store and the Forward
- Integration and Transformation Layer (ITL): Transforms the data from the Staging database format to the target database format

Collaboration:

- Triggered by human control, timer or other applications, the data pumps take a copy of the source systems and send them to the Delta
- The Delta identifies changed data from the last load of data and marks the Staging DB with data to be changed at the target location

Triggered by human control, timer or other applications, the Integration and Transformation Layer parses the marked data following some rules and sends the results to the ODS

8. **Proof of Concept : GEDIS**

An implementation of GODIS has been made in collaboration of CGI and CRCD-RDDR Valcartier. Generic Enterprise Data Integration Solution (GEDIS) is the resulting framework of that project that can be applied directly into enterprise in order to build ODS or resolve some data integration problems.

GEDIS is essentially a framework composed of component tools. Some of the components are ready to be used, some need to be configured; others need to be adapted to the context of the enterprise. In every case, GEDIS is not an out of the box product. GEDIS is an adaptive solution that must be extended for each enterprise that wants to use it. The next diagram is the hi-level view of GEDIS.

Also, GEDIS doesn't provide the capability to do application integration, even though the data collected in the ODS can be used by some operational systems to avoid having to link with other operational systems for data. GEDIS helps enterprise to aggregate the information, not the functionality.

To do this, GEDIS uses seven components, divided into two major layers. The first layer is the Staging, the second one is the Integration and Transformation. Between each layer, the data stops in the middle database; the Staging database. The processing of each layer is independent and the components are designed to work concurrently.

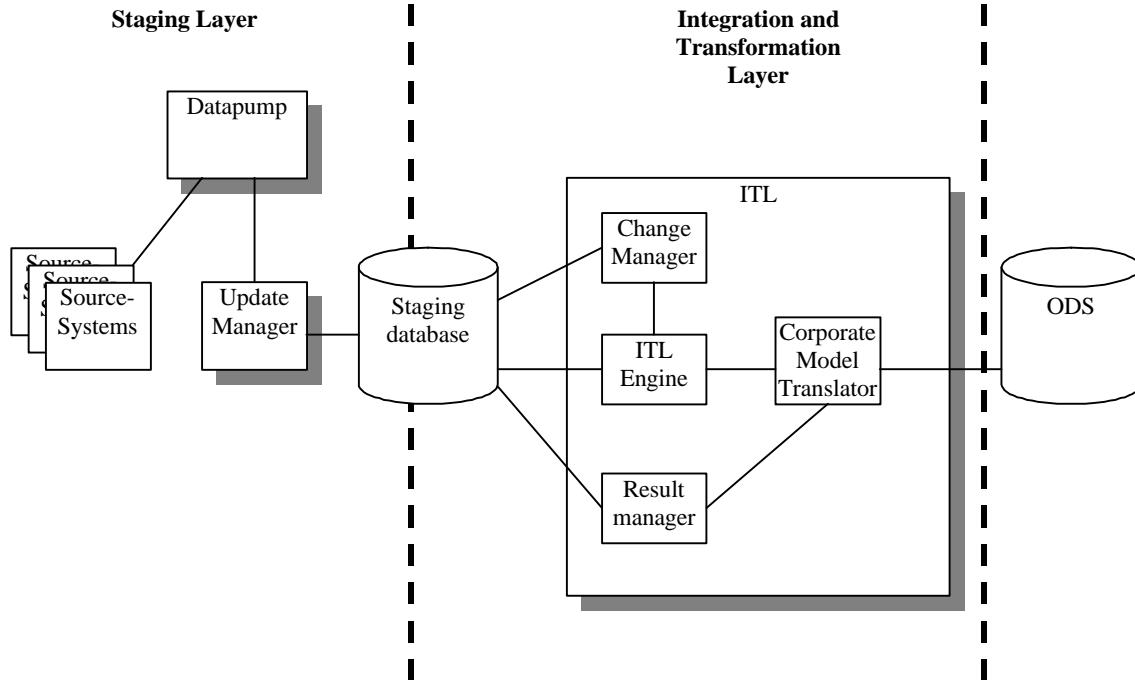


Figure 7 General architecture

8.1 GEDIS Components framework

This grid gives short explanation of the component.

Component	Sub-Components	Implemented	Roles and responsibilities
Datapump	-	Yes	The datapump is responsible for extracting source-system information in various sources and transforming it into XML documents.
Update manager	-	Yes	The update manager is responsible for keeping a local copy of source system data in the staging database. It is also responsible for determining which one has change since the last verification. It changes the data status in the staging database.
Staging database	-	Yes, schemas	The staging database contains the last image of the legacy system that is waiting to be processed. Also, the staging database contains information for integration, transformation and configuration of the system.

ITL	Change manager	Yes	Change manager is responsible for taking the marked data to be send to the ODS. Also, it is responsible for insuring the synchronisation between update manager and ITL Engine.
	ITL Engine	Yes, may need extensions for specific processing contexts	Responsible for transforming and integrating data
	Corporate model translator (CMT)	Yes, need to be extended to specific ODS database technologies	Responsible for pushing the data into the ODS. The general frame of CMT is developed; it needs to be extended to specific ODS technologies. (SQL, CORBA, XML, SOAP, DCOM ...)
	Result manager	Yes	Responsible for logging the execution status of the ITL process. That component is located at the end of the ITL layer process and perform the synchronization of the data after they have been touched by the CTM. It take data in Entities object adjust the corresponding row in the staging database.
ODS	-	No	Operational Data Store database, used to keep integrated and transformed data on Corporate Model for analytic applications.

8.2 *Communication between components*

As the two layers of GEDIS are independent, communication within GEDIS can be divided in two parts corresponding to the layers. And as the GEDIS framework is modular, there is a lot of way in which modules share information. Those are the technologies used for communication within GEDIS:

- XML: Some of the communication uses streamed XML to ensure a portable ways to exchange data from a components to another one when component are not in the same running environment.

- SQL-92: The query standard is used to communicate between staging database and components. This standard ensures that every SQL-92 database can be used for staging.
- Entity buffer: When GEDIS component are in the same running environment, the best way to exchange information is memory pointer exchange. The entity buffer is used to do the pointer exchange. This is a FIFO buffer that works only with a specific object type to have a maximum of performance.

The following figure describes communications within Staging Layer:

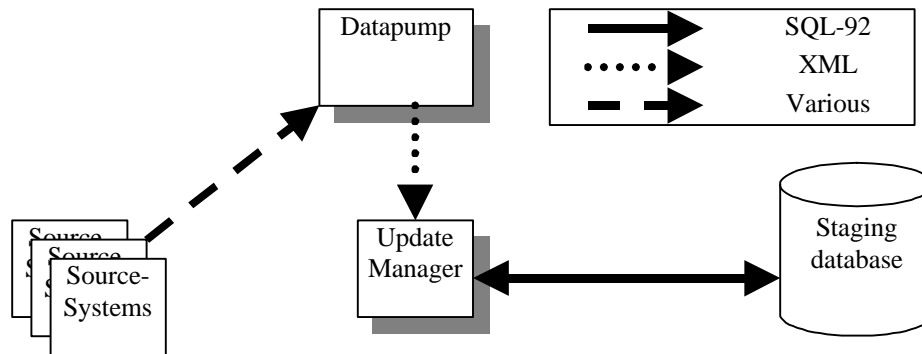


Figure 8 Staging Layer Data Exchange

This grid explains communication between the staging layer components.

From	To	Bi-directional	Type	Content
Source-system	Datapump	No	Various	The datapump performs some queries on various types of source data (Oracle, MainFrame, Acces, flat file, Excel, XML, etc)
DataPump	Update Manager	No	XML	XML that respects a specific frame is sent over network or written in XML files on disk.
Update Manager	Staging Database	Yes	SQL-92	Use SQL-92 queries to save data and their statuses.

The following figure describes communications within the integration and transformation Layer:

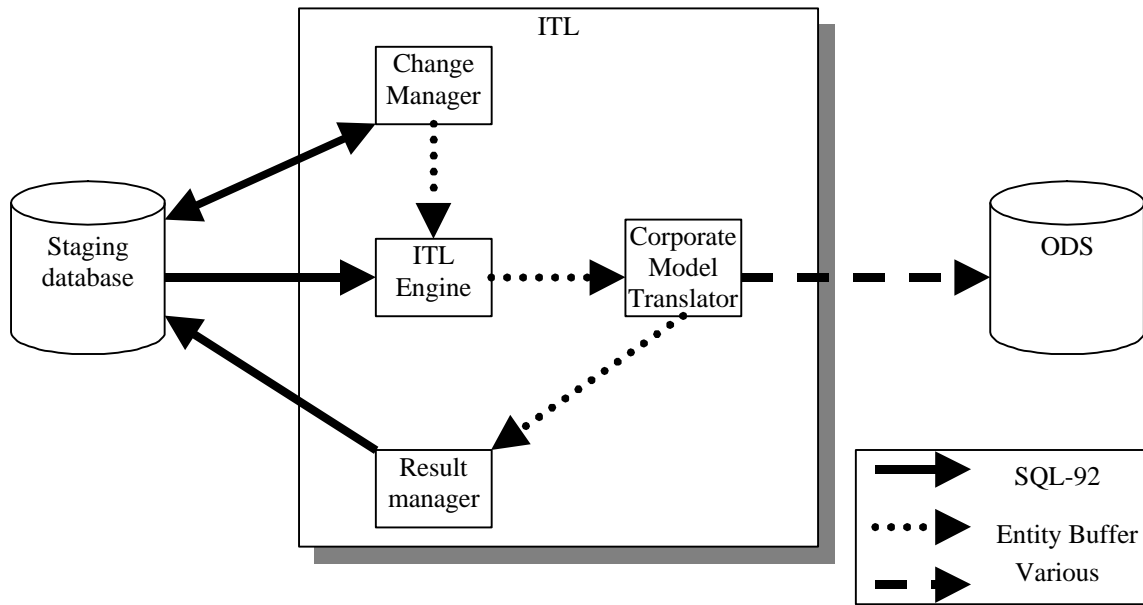


Figure 9 - ITL Data Exchange

This grid explains communication into the integration and transformation layer.

From	To	Bi-directional	Type	Content
Stating database	Change Manager	Yes	SQL-92	Data in the staging database are marked as “in process” and then are put in memory.
Change Manager	ITL Engine	No	Entity Buffer	The Change Manager puts newly retrieved entities form the database into the Entity Buffer for waiting to transformation by the ITL engine.
ITL Engine	Corporate Model Translator	No	Entity Buffer	The ITL Engine puts the transformed entities into the Entity Buffer to waiting for the processing by the corporate model translator.
Corporate Model	Result Manager	No	Entity Buffer	The CMT puts sent entities in the Entity

Translator				Buffer to be synchronised within the staging database by the result manager
Corporate Model Translator	ODS	No	Various	The way that the CMT transfer information to the ODS depends in the specific implementation of the Corporate Model Translator.
Staging database	ITL Engine	No	SQL-92	The ITL Engine takes information about integration and transformation from the staging database.
Result Manager	Staging database	No	SQL-92	The Result Manager puts the entities in staging in their right final status and puts error information in the database if necessary.

8.3 *Multi-threading*

The loading of data from the source system to the ODS is designed to process via multi-threading. The first layer can work independently from the second layer. The staging database and the verification of the status of components ensures the synchronisation. You can start multiple instances of the staging layer or the integration and transformation layer simultaneously.

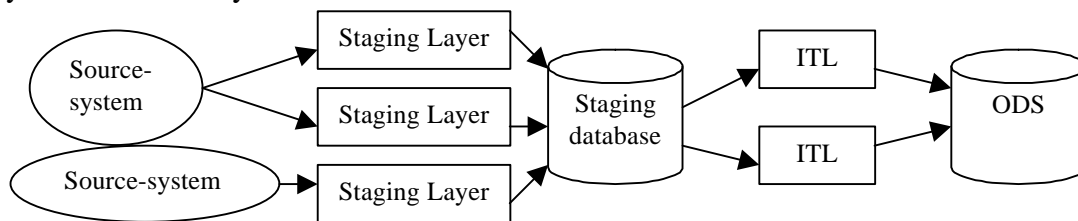


Figure 10 Multi-Threading within Layers

8.4 *Control console and distant listener*

To remotely control the various components of both the Staging Layer and Integration and Transformation Layer, GEDIS is provided with a set of components. The three components of GEDIS, datapump, updateManager and ITL, are already implemented to be used in that remote control context.

The remote control consists in a client side that receives orders over the TCP/IP framework and it is called the “Executable Listener”. The server side is a component that you will link to the user interface called the Executable. In the Model-View-Controller design pattern point of view, both client and server parts provide the Model part to link with the user interface of the system. As shown in the next figure, GEDIS is designed for being integrated into a Model-View-Controller design pattern [Gamma and al. 1994].

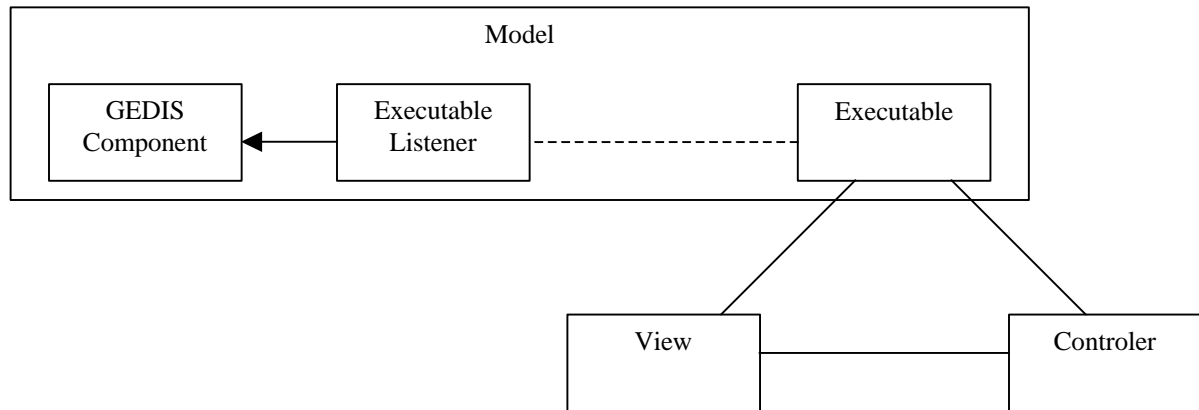


Figure 11 Model View Controller within GEDIS

GEDIS comes with a simple user interface that provides an implementation reference and an example of the use of the exesvc package. If you need to provide in your implementation of GEDIS a very simple user interface, the provided interface will allow you to control the GEDIS flow easily. That reference implementation is called the console. If your implementation of GEDIS needs complex operations like scheduling and automatic error recovery, the software builds over the existing remote control framework could eventually embed these features.

Even though all the components are designed to work in the remote control environment, it's possible to start all the components of GEDIS in a stand-alone mode.

9. Conclusion

Even this solution could not respond to all the needs of coalition interoperability but it's a good step forward in order to enable different countries to have a unified view of a situation. The GEDIS experience shows that this solution could be implemented, perform an efficient merge of information in a short timeframe and enable armies to use normally their current C2 software while having a coalition level C2 interoperability.

The recent war experiences show that the coalitions are assembling themselves faster than ever and that they often include many unexpected players. While waiting for a complete standardisation process of the of the NATO forces, this solution can help forces to work together.

10. Bibliography

[Inmon, 1995] Inmon W H. Building the operational data store. John Wiley, NY, NY 1995

[Inmon, 1998] Inmon W H. Corporate Information Factory. John Wiley, NY, NY 1998

[Gamma and al. 1994] Gamma and al. Design Patterns Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company. Reading, MA 1994