

Data Architecture



DATA ARCHITECTURE

GITS describes Data Architecture as a discipline that is predicated upon developing a knowledge-base associated with the usage and management of information. It is comprised of the following components:

- Data Profiling
- Identification Of Data Abstractions And Structures
- Data Warehouse/Data Mart Design
- Business Processes/Business Rule Management
- Mapping of Data Abstractions/Structures to Application Objects
- Data Integration Methods
- Quality Control Rule Definition
- Support of Security Entitlements

DATA PROFILING

As data is acquired, GITS provides guidance on how to use data profiling as input during the modeling process. GITS provides a cohesive set of steps and guidelines pertaining to the profiling of data. We will help your enterprise determine the following:

DATA CONFORMANCE

We will help you to ascertain and properly document the patterns of data, data types, representative values and statistics associated with the data for each Data source. GITS uses the CBL/Taxonomy, along with values captured during data profiling to ensure that incoming data is consistent with what is captured in the Meta Data repository and underlying data structures.

ENTERPRISE USAGE

GITS will show your application teams how to share information with other application areas without upsetting their project timelines.

RISK MANAGEMENT

We will work with your enterprise and provide you with Best Practices to determine the risks associated with the using data from a given Data Source.

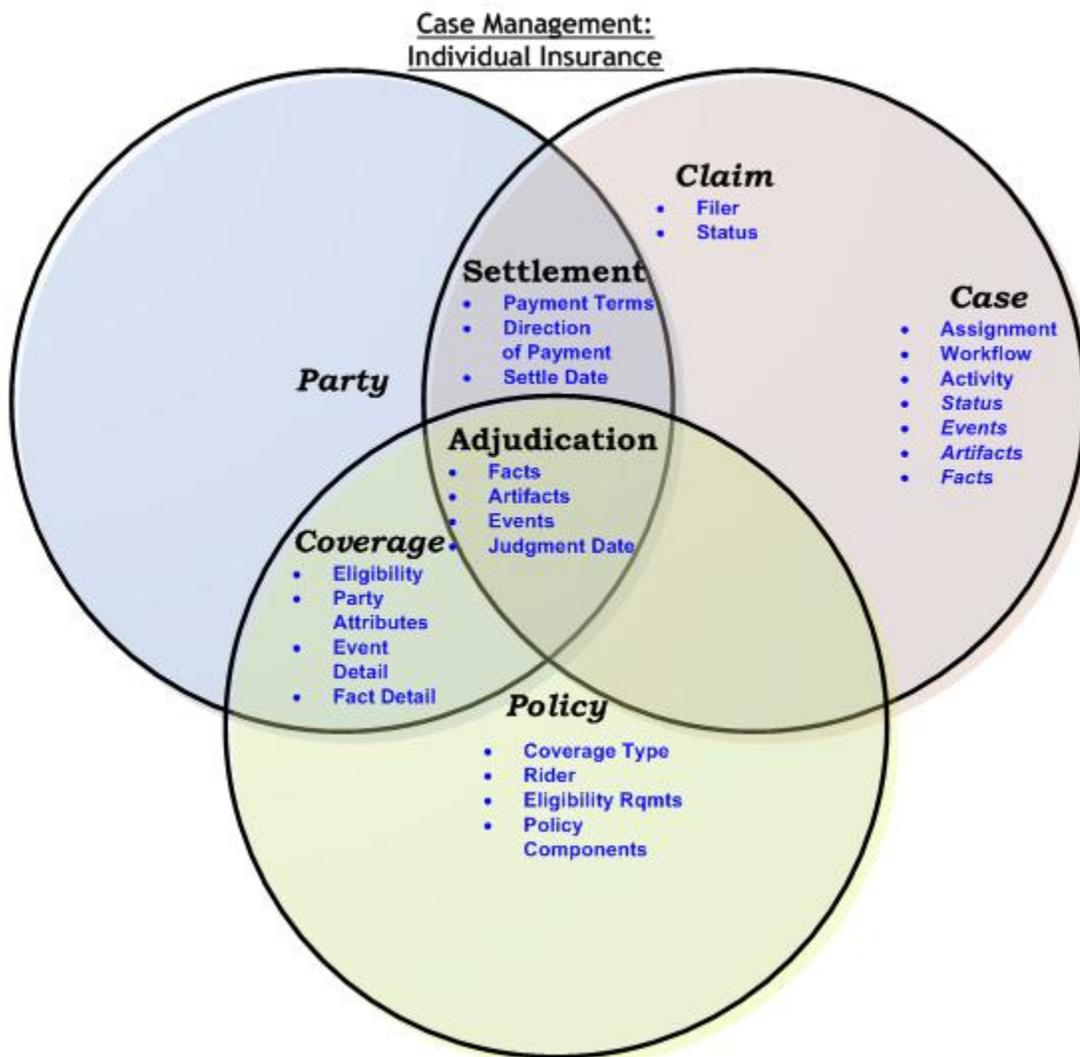
INTERDEPENDENCIES

Data interdependencies occur when there is a mutually inclusive relationship between the values contained in two data items. For instance, the values contained within Postal Code (e.g., Zip Code in the U.S.) are tightly coupled with the values contained within the City Code

IDENTIFICATION OF DATA ABSTRACTIONS AND STRUCTURES

It is not unusual for end users, managers and application developers to be unfamiliar with entity relationship diagrams. GITS understands this and also recognizes that their participation is mandatory for the successful deployment and development of a solution.

Therefore, we have developed CBL/Taxonomy Palettes that can be used in conjunction with the data models.





DATA WAREHOUSE/DATA MART DESIGN

The semantic data model as depicted in the previous page is used as a springboard for creating data warehouse and data mart designs. GITS has developed the following database designs in quite a few Industry segments:

- Decision support models (3nf, de-normalized structures, etc.)
- Analytical processing models (OLAP, ROLAP, MOLAP, etc.)

BUSINESS PROCESSES/BUSINESS RULE MANAGEMENT

The resources at GITS have used a wide variety of methodologies to model business processes. GITS has a data driven method of capturing the logic associated with business processes. Our methodology allows one to encapsulate these rules and keep them separate from ETL logic. The fact that the methodology is data driven, the amount of coding necessary to maintain the structures is minimal.

MAPPING OF DATA ABSTRACTIONS/STRUCTURES TO APPLICATION OBJECTS

GITS believes that it is important to build a catalog that contains a cross-sectional application usage of data objects. We provide our clients with an off the shelf repository expressly designed for this purpose.

DATA INTEGRATION METHODS

GITS provides a set of mapping templates and pseudo code that can be used to streamline the data integration process. The CBL/Taxonomy is used to ensure that data items received from various data sources all aim at a common target.

QUALITY CONTROL RULE DEFINITION

Incoming data is rarely perfect all of the time. There may be instances in which the data may need to be enriched. GITS has provided its clients with a repository that can be used to document and enforce these rules.

SUPPORT OF SECURITY ENTITLEMENTS

GITS understands the importance of protecting your information. Our resources have worked closely with application developers in providing a data driven approach to managing security entitlement rules.