

## Il linguaggio GDL

Il GDL<sup>1</sup> è un linguaggio di programmazione parametrica definito dalla software-house Graphisoft all'inizio degli anni '80 per lo sviluppo, anche da parte di terzi, di librerie di oggetti parametrici da utilizzare nel proprio software architettonico Archicad. Le peculiarità di questo linguaggio di modellazione sono notevoli, permette infatti, tramite piccoli script in formato testo, la creazione di qualsiasi tipologia di oggetto architettonico e non, tridimensionale o bidimensionale, completamente parametrico e quindi riconfigurabile nelle proprie caratteristiche dimensionali, tramite una finestra di dialogo, in qualsiasi momento dall'utente finale.

L'estrema *leggerezza* del file sorgente, espresso in formato testo, in termini di dimensioni in KB<sup>2</sup> si traduce in una notevole facilità di archiviazione e trasmettibilità dei dati; ci basti pensare che un oggetto in GDL può *pesare* 10 KB mentre il suo corrispettivo in altri formati 1 MB o più, un rapporto a favore di uno a cento.

E' inoltre da tenere presente che l'immagine tridimensionale dello script è ottenuta tramite un traduttore interno al software Archicad, ora disponibile come plug-in<sup>3</sup> per altri software CAD come AutoCAD.

Il file di disegno è quindi ottenuto attraverso una sequenza di funzioni matematiche e coordinate cartesiane entrambe lette ed interpretate,

---

<sup>1</sup> Geometric Description Language

<sup>2</sup> Unità di misura della quantità di memoria necessaria per l'immagazzinamento dei dati.

8 bit = 1 byte  
1024 byte = 1 KB  
1024 KB = 1 MB  
1024 MB = 1GB

<sup>3</sup> Programma che si inserisce su un altro software, per fornirgli delle funzionalità nuove, ma che senza di esso non esplica alcuna funzione.

volta per volta, dal software per generare le operazioni di calcolo richieste nelle differenti visualizzazioni. Le primitive, che compongono il modello 3D, vengono create nello spazio tridimensionale misurato secondo gli assi del sistema di coordinate principale x, y, e z.

L'origine di questo sistema è il punto (0,0,0) rappresentante l'origine assoluta del sistema cartesiano degli assi ed ogni indicazione, contenuta negli script GDL si basa su questo sistema di coordinate. L'editazione di uno script GDL avviene tramite l'apposito modulo interno ad Archicad, al quale si accede dal menù Archivio -> crea nuovo elemento di libreria, oppure tramite altri software stand-alone.

E' strutturato in varie parti, singolarmente compilabili, che qui di seguito esponiamo brevemente.

- **Testo Master**

Contiene le istruzioni che vengono interpretate tutte le volte che viene chiamato lo script, è quindi opportuno inserirvi le condizioni chiave del nostro oggetto, siano esse parametriche oppure no.

- **Testo GDL 2D**

E' lo spazio di editazione della descrizione bidimensionale dell'oggetto creato; che può essere bidimensionale, tridimensionale o entrambi contemporaneamente (2D+3D). Nel caso di una realizzazione in 3D, il **testo GDL 2D** conterrà solamente le istruzioni per la rappresentazione in pianta dell'oggetto.

- **Testo GDL 3D**

Contiene le descrizioni della forma tridimensionale dell'oggetto (se prevista), possiamo definirlo insieme al testo 2D la parte fondamentale dello script.

### **Testo Attributi**

Vi si possono definire le macro degli attributi e collegare descrizioni e/o dati tecnici dell'oggetto.

### **Testo Parametri**

Vi si possono impostare una o più liste di valori dei parametri che controllano la geometria e le proprietà dell'oggetto.

- **Testo Interfaccia**

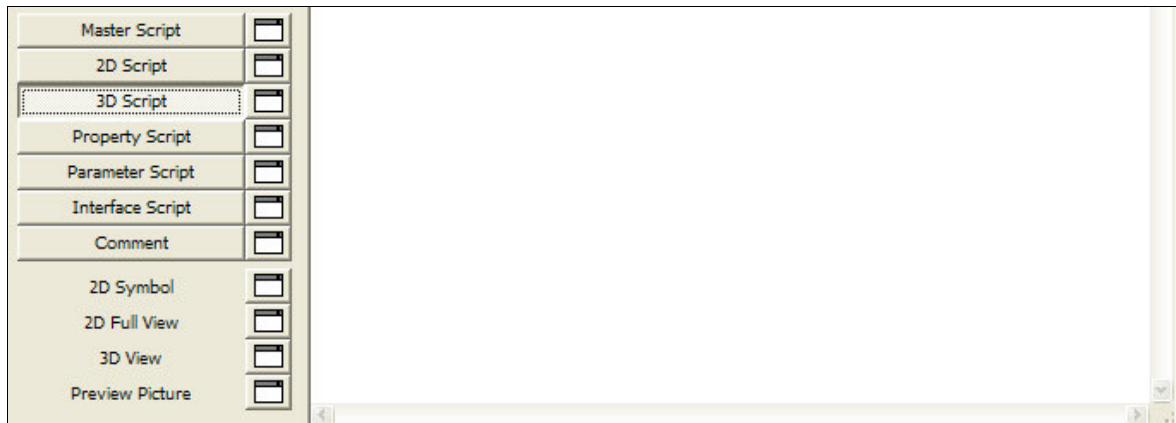
Gestibile solo a partire dalla versione 6.5 del software Archicad, dà la possibilità di personalizzare la finestra dei parametri che ne guida l'inserimento nel foglio di lavoro.

- **Commento**

Vi si possono inserire note tipo la data di creazione dell'oggetto il nome del programmatore, eventuali propositi di miglioria etc.

- **Finestra inserimento parametri**

Vi si inseriscono i parametri di default che descrivono il modello



**Figura 1: Finestra di editazione degli script GDL all'interno di Archicad**

Variabile	Tipo	Nome	Valore
A		Dimensione X	1,0000
B		Dimensione Y	1,0000
<b>Definizione 3D</b>			
altezza		H pilastro	2,0000
altezza2		H cono	1,0000
r_cilindro		R cilindro	0,3000
r_base		R cono	0,3000
r_superiore		R cono sopra	0,5000
<b>Materiali</b>			
mat1		materiale cilindro	1
mat2		materiale cono	4
<b>Definizione 2D</b>			
cpn		contorno base pilastr	1
cft		retino base	5
cfp		penna retino	1

Figura 2: Finestra di inserimento parametri di default all'interno di Archicad

E' opportuno ricordare che per il corretto funzionamento dello script, sono necessari il **testo GDL 3D**, se operiamo esclusivamente nel tridimensionale, il **testo GDL 2D**, se creiamo oggetti solamente bidimensionali, ed ovviamente entrambi per oggetti completi, le altre parti sopra elencate, contribuiscono esclusivamente ad una maggiore "pulizia" ed ottimizzazione del codice dello script.

Il GDL, come già ampiamente detto, è un linguaggio di programmazione. Tale definizione ha creato una sorta di ghetizzazione del suo utilizzo, credendo i più che sia cosa riservata esclusivamente agli informatici puri. Questo è un grande errore, in realtà il linguaggio GDL è stato pensato per gli utilizzatori del software Archicad, si rivolge quindi a tutti quegli operatori, architetti, ingegneri, geometri che abbiano la volontà di imparare alcune regole sintattiche proprie del linguaggio. Paradossalmente le conoscenze fondamentali richieste per un corretto

utilizzo del GDL, sono quelle della matematica di base, della geometria trigonometrica e dello spazio cartesiano. In pratica è necessaria la capacità di saper pensare le forme geometriche per quello che sono realmente: *espressioni numeriche*. Dimostrazione di ciò è la sintassi che si utilizza per redazione di uno script, essa è relativamente semplice e richiama chiaramente il significato di ciò che stiamo scrivendo, così ad esempio il comando CYRCLE2 indica che stiamo costruendo un cerchio nel sistema bidimensionale, o con raggio di valore numero reale finito -CYRCLE2 10-, oppure di valore variabile -CYRCLE2 rag- in cui la denominazione "rag" definisce direttamente la variabile come vedremo più avanti. Ovviamente questo è un esempio estremamente semplice, nel caso di forme complesse si è chiamati a ben altre scritture che spesso tendono a rivelarsi, ad un occhio profano, indecifrabili.

E' comunque vero, come ben sanno tutti coloro che si sono cimentati con la modellazione tridimensionale, che non esiste, e probabilmente non esisterà mai, un software *user friendly* tale da permettere a chiunque di ottenere fin da subito i risultati sperati, senza dover affrontare una difficile e spesso noiosa *curva di apprendimento*. Ci si può allora chiedere come mai il GDL sia un strumento di nicchia, la risposta è molto semplice e senza voler entrare in particolari analisi sulle diverse scelte di mercato che le varie software-house hanno compiuto negli anni, è evidente che la mancanza di una interfaccia grafica, da poter utilizzare per la redazione degli script, rappresenta un handicap che molti reputano insormontabile (per la costruzione di un cerchio è molto più immediato cliccare sull'icona che lo raffigura piuttosto che ricordarsi di digitare CYRCLE2).

Queste considerazioni sinora fatte non devono comunque trarre in inganno sulle reali potenzialità di questo linguaggio, è vero che è uno strumento di disegno e/o modellazione, ma queste non sono che le sue funzionalità più esplicite. In realtà come già spiegato precedentemente, in uno script GDL possono essere editate tutta una serie di informazioni riguardanti l'oggetto in questione, che nessun altro software è capace di fornire e permettono al progettista una visione in toto del suo lavoro, passando dalla quantificazione dei materiali necessari per la sua realizzazione, sino ad una verifica strutturale di massima.

Facciamo ora un semplicissimo esempio, supponendo di voler progettare un pilastro circolare soggetto a carico concentrato verticale non eccentrico, i comandi nel **testo GDL 3D** saranno:

```
mat M  
CYLIND H, Rag
```

dove la variabile "M" rappresenta il tipo di materiale, la variabile "H" definisce l'altezza del pilastro e la variabile "Rag" il suo raggio.

A questo punto il pilastro è costruito nello spazio con centro coincidente con l'origine dagli assi cartesiani in base ai valori numerici (H e Rag) e distintivi (M) attribuiti alle tre variabili in fase di collocazione sul disegno.

Dovremo poi preferibilmente nel **testo master**, definire la verifica di massima per il tipo di sollecitazione a cui è soggetto il pilastro, per cui scriveremo:

```
Area=pi*Rag^2
sigma=S1
carico=N
sollecitazione=(carico/Area)
IF sigma<sollecitazione THEN
PRINT "Attenzione il pilastro non è verificato" ,sollecitazione
ENDIF
```

Dove "Area" è la definizione della sezione resistente in funzione della variabile "Rag", "S1" è il valore della sigma in funzione del materiale, "N" è il valore del carico concentrato, entrambi impostabili dall'utente in fase di utilizzo dell'oggetto pilastro nel file di disegno, "sollecitazione" è il risultato della formula per carichi normali concentrati.

Le ultime tre righe permettono di definire le condizioni per le quali vogliamo ricevere un avviso, tramite finestra di dialogo testuale, sull'esito positivo o negativo della verifica di massima.

Al momento del suo utilizzo il progettista, scegliendo lo script del pilastro dalla propria libreria di oggetti, dovrà solamente deciderne le dimensioni e il materiale, indicare il valore del carico e attendere pochi istanti per ottenere il risultato della verifica, e quindi decidere quale sia la migliore soluzione che soddisfi forma e struttura.

Si fa notare che con la scrittura sopra esposta, anche in caso di verifica negativa del pilastro, tale oggetto verrebbe comunque correttamente posizionato nel file di disegno, non inficiandone assolutamente l'utilizzo. Possiamo scrivere invece condizioni più drastiche che in caso di verifica negativa blocchino lo script, non permettendo di fatto il suo posizionamento nel disegno, oppure che adeguino le dimensioni

dell'oggetto al carico impostato per ottenere sempre una verifica positiva.

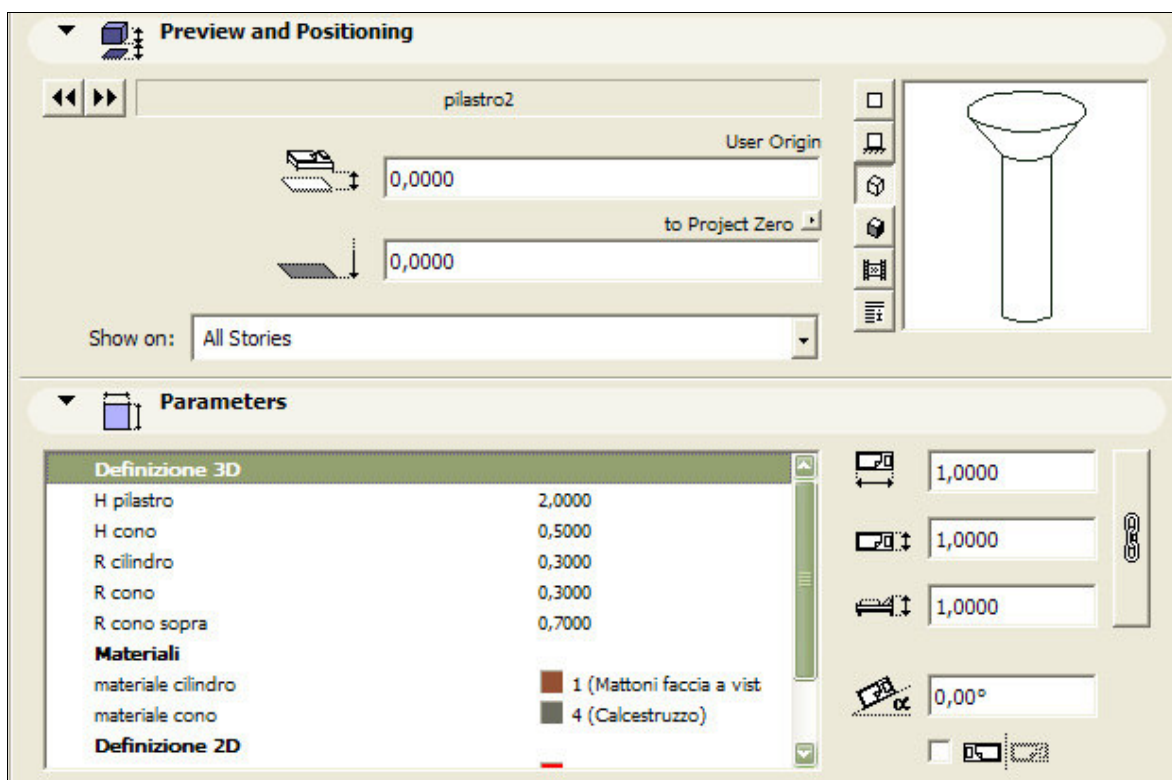


Figura 3: Finestra di posizionamento dell'oggetto in Archicad, in questo esempio è stato aggiunto un cono alla sommità del cilindro.