Dynamo: la replicazione automatica dei nodi

Nella creazione di script in Dynamo spesso ci si trova a gestire non un singolo dato ma una **serie di dati**. aggregati in apposite strutture dette liste. Le liste possono essere a loro volta composte da altre liste e questo concetto può essere ripetuto un numero n di volte **aumentando la profondità** della lista. La misura della profondità della lista è detta rango R e vale 0 per valori singoli, 1 per liste composte da singoli elementi, 2 per liste di liste e via dicendo.

Quando si fornisce come input ad un nodo una lista di rango n, ci si aspetta che Dynamo applichi ripetutamente la funzione del nodo a tutti gli elementi della lista e di fatto questo è ciò che realmente avviene. Tale meccanismo si chiama **replicazione automatica** ed è controllato da delle regole ben precise non sempre intuitive.

Applicazione della replicazione automatica al nodo SetParameterByName

Facciamo un esempio su un caso pratico e abbastanza frequente: a partire da dei dati tabulati vogliamo utilizzare il nodo "**SetParameterByName**" per compilare le proprietà di una serie oggetti di Revit. Assumendo che le righe della tabella corrispondano in numero agli oggetti di Revit, l'abbinamento tra oggetto e riga della tabella avviene attraverso il campo Id, che funge quindi da chiave. Le proprietà da compilare sono 4 valori di una WBS fittizia (Tabella 1).

Id	Livello	Famiglia	Tipo	WBS_L0	WBS_L1	WBS_L2	WBS_L3
136127	L_00	Doppia Anta	Doppia Anta	C1	E1	L00	Do
136131	L_00	Doppia Anta	Doppia Anta	C1	E1	L00	Do
136135	L_00	Doppia Anta	Doppia Anta	C1	E1	L00	Do
170922	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
171617	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
171706	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
171711	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
171716	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
171721	L_00	Finestra - 1 Anta	100x240 cm	C1	E1	L00	Wi
179687	L_00	Finestra - 1 Anta	100x100 cm	C1	E1	L00	Wi
179766	L_00	Finestra - 1 Anta	100x100 cm	C1	E1	L00	Wi
144639	L_00	Floors	Latero Cementizio - 30 cm	C1	E1	L00	Fl
192598	L_00	Floors	Calcestruzzo da 160 mm con soletta metallica da 50 mm	C1	E1	L00	Fl
183727	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
183728	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
183729	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
183730	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
184070	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
184091	L_00	Porta - 1 Anta	80x210 cm	C1	E1	L00	Do
133339	L_00	Walls	GP_Generico_pannelli chiari	C1	E1	L00	Wa
134087	L_00	Walls	GP_Generico_pannelli chiari	C1	E1	L00	Wa
134325	L_00	Walls	GP_Generico_pannelli chiari	C1	E1	L00	Wa
134587	L_00	Walls	GP_Generico_pannelli chiari	C1	E1	L00	Wa
134659	L_00	Walls	GP_Generico_pannelli chiari	C1	E1	L00	Wa
134992	L_00	Walls	GP_Generico_pannelli scuri	C1	E1	L00	Wa
135371	L_00	Walls	GP_Generico_pannelli scuri	C1	E1	L00	Wa
135704	L_00	Walls	Generico - 30 cm	C1	E1	L00	Wa
135883	L_00	Walls	GP_Generico_Vetrata	C1	E1	L00	Wa

Tabella 1

Importata dal file CSV tramite il nodo "Data.ImportCSV" e limitatamente alle 4 proprietà in esame, la tabella appare come **una lista di rango 2** (Figura 1), composta da 211 liste (una per ogni riga della tabella) di 4 valori ciascuna, uno per ogni proprietà da compilare. La lista delle righe è più esterna rispetto a quelle dei valori.

L'importazione di una tabella CSV in Dynamo genera sempre una lista di righe di dati.

Utilizzando ad esempio il nodo "**All Element of Category**" e lavorando opportunamente sui dati si estrae l'elenco degli oggetti di Revit a cui assegnare le proprietà (Figura 2), mentre dalle intestazioni di colonna i nomi delle proprietà da valorizzare (Figura 3).

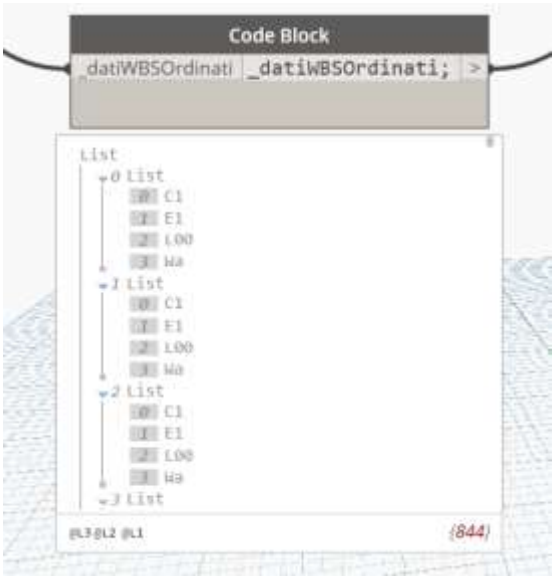

Figura 1

Figura 2

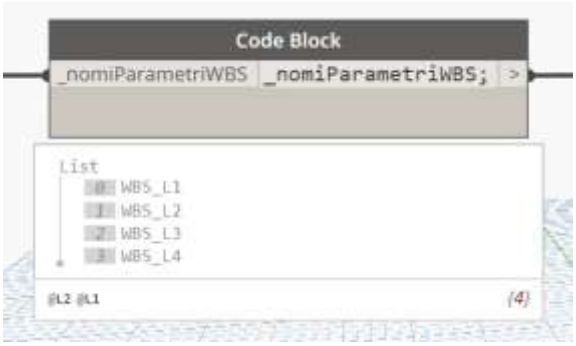

Figura 3

Applicando il nodo **SetParameterByName** non si ottiene però il risultato atteso. La replicazione avviene ma non nel modo corretto: è evidente considerando l'output del nodo: 4 parametri per solo 4 elementi invece dei 211 elementi attesi (Figura 4).

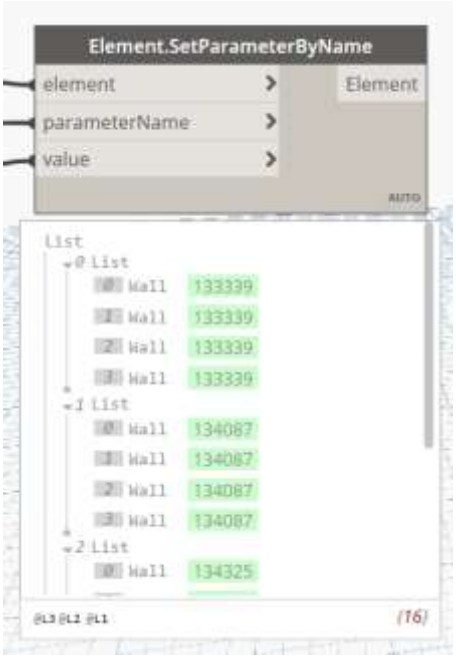

Figura 4

Parametri e argomenti

Da un lato c'è il nodo che si aspetta in input un **parametro** di un certo rango Rp, dall'altro ci sono gli input che lo script gli fornisce, cioè gli **argomenti**, di rango Ra. Il rango dei parametri è desumibile dalla **descrizione del nodo**: ad esempio "List.MaximumItem" (Figura 5) si aspetta in input una lista formata da elementi di qualsiasi tipo, quindi rango 1, e ciò si indica con var[].

Figura 5

In generale si possono avere i seguenti casi:

dimensione lista = **rango R (rank)**

var	valore singolo	R=0
var[]	lista mono dimensionale	R=1
var[][]	lista bidimensionale	R=2
var[][][]	lista tridimensionale	R=3
var[]..[]	lista n-dimensionale	R=n

Per ciascun input non è detto che il rango del parametro Rp corrisponda a quello dell'argomento Ra. Se il nodo si aspetta un parametro di rango 1 e noi forniamo un argomento di rango 3, Dynamo deve **replicare la funzion**e in modo da iterare su tutti i valori delle liste. È come se ogni parametro avesse una certa capienza Rp superata la quale avviene la replicazione del nodo e l'iterazione degli elementi nelle liste. Ma come avviene la replicazione?

Le regole della replicazione

Per ogni parametro i-esimo si valuta la differenza **dRi** tra il rango dell'argomento fornito e quello del parametro atteso, tralasciando i parametri con rango generico (var[]..[]). Dopodiché:

- Se in almeno due casi si ha un dRi maggiore di zero, si applica l'iterazione **zip breve** sui valori di tali argomenti, cioè la funzione del nodo viene ripetuta accoppiando uno a uno gli elementi delle liste più esterne troncando il risultato sulla lunghezza della lista più corta. Gli altri argomenti che non necessitano di iterazione rimangono fissi all'interno di ciascuna ripetizione del nodo.
- Se in un solo caso si ha dRi maggiore di zero, si applica la ripetizione **cartesiana** su tale argomento, cioè la funzione del nodo viene ripetuta sugli elementi della lista più esterna mentre gli altri argomenti che non necessitano di ripetizione rimangono fissi.
- Si sottrae 1 da ciascuno dei dRi precedentemente calcolati e si riparte da capo, fermandosi quando tutti i dRi sono uguali a zero.

Dalla procedura sopra descritta emerge che:

- la replicazione può avvenire diverse volte, che identificheremo come step, fino a quando tutti i dRi non si annullano
- in caso di replicazione zip breve tra liste di lunghezza diversa, i valori eccedenti della lista più lunga vengono troncati e quindi non entrano nella valutazione del nodo
- i diversi livelli di replicazione agiscono dall'esterno verso l'interno delle liste e ad ogni passaggio si va ad intaccare un livello più profondo (ecco perché si sottrae 1 dai dRi)

Ora è possibile spiegare cosa è successo con il nodo SetParameter secondo le regole della replicazione.

Dalla firma del nodo (Figura 6) si vede che tutti i parametri hanno rango 0 essendo elementi singoli di tipo Element, string e var rispettivamente.

Gli argomenti forniti hanno invece, nell'ordine, rango 1, 1 e 2. Il valore del rango è simbolicamente rappresentato dai quadrati colorati (Figura 7)

Figura 6

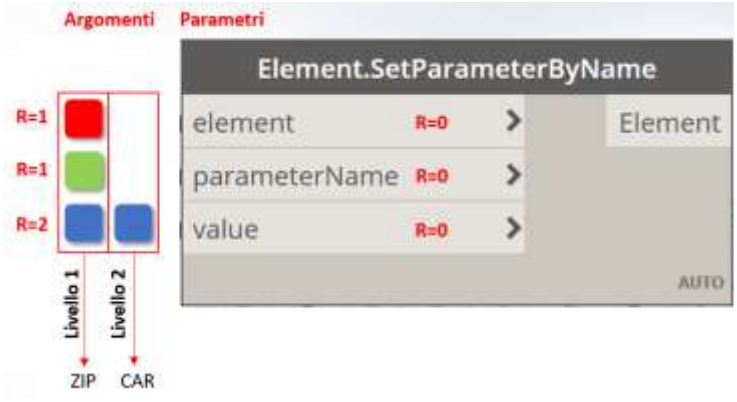

Figura 7

STEP 1: replicazione zip breve tra i 3 argomenti. La corrispondenza avviene a livello più esterno delle liste. Si noti che essendo la lista dei nomi dei parametri quella più corta e composta da 4 elementi, quelle degli oggetti di Revit e dei valori dei parametri verranno troncate a tale lunghezza (Figura 8). Inoltre la corrispondenza uno a uno dei membri delle liste comporta che nel primo oggetto Revit verrà compilata solo la proprietà WBS_L1, nel secondo solo WBS_L2 e così via (Figura 9). L'elemento utilizzato per valorizzare la proprietà in ciascuno dei 4 casi sarà una lista, non un singolo valore come il nodo si attende, quindi avverrà una seconda replicazione.

Allo step 1 il nodo viene quindi ripetuto in totale 4 volte.

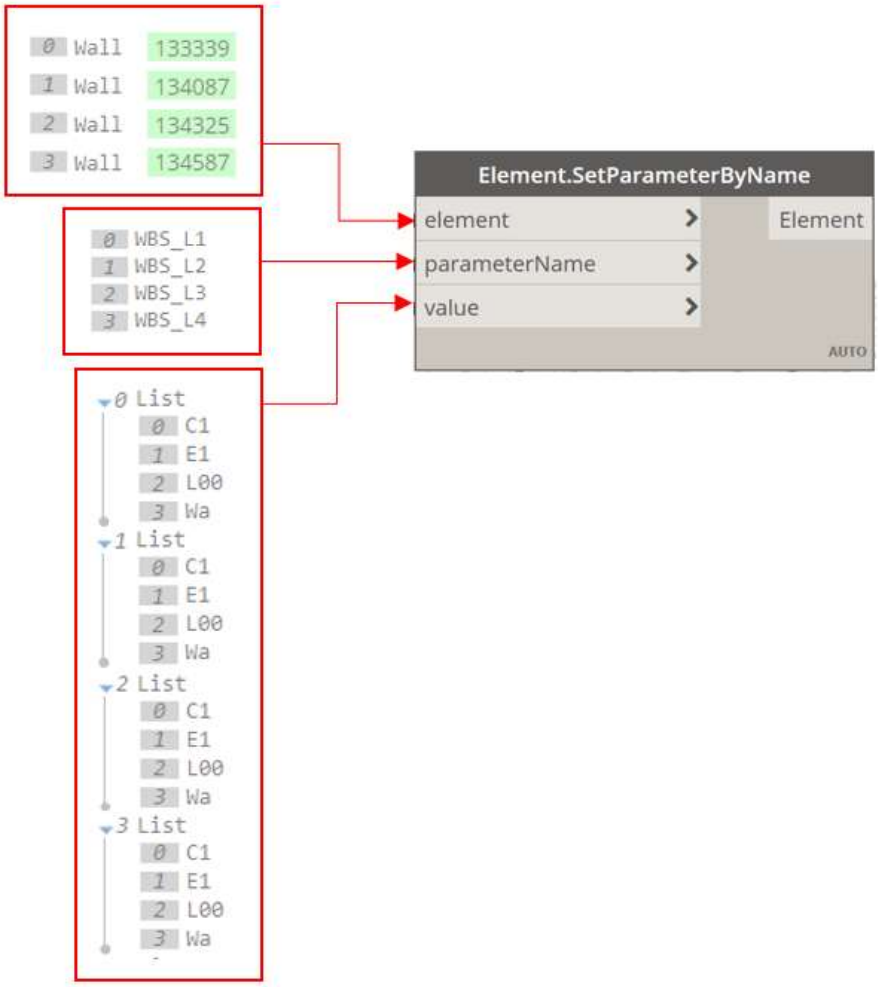

Figura 8

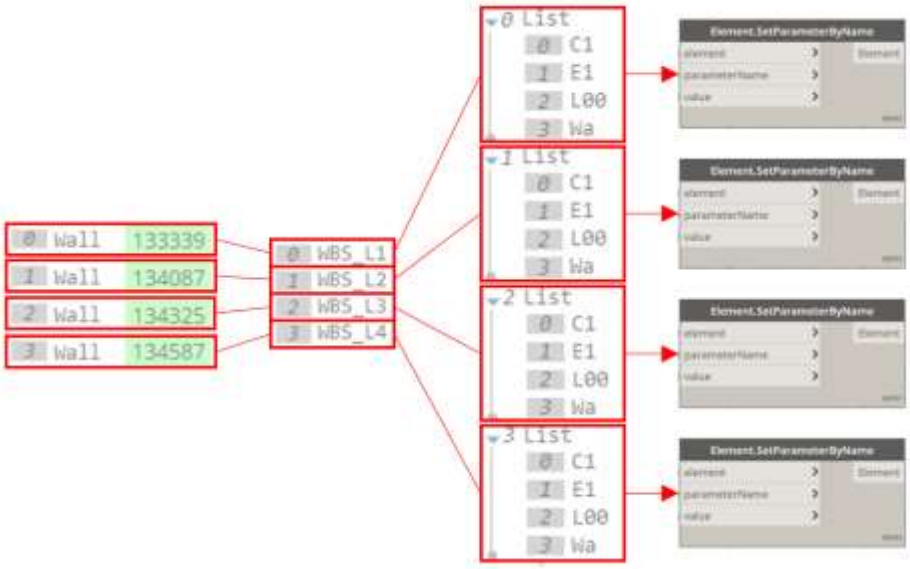

Figura 9

STEP 2: replicazione cartesiana. Gli argomenti element e parameterName non richiedono più iterazione in quanto sono singoli valori con rango 0 per i quali dR=0. L'argomento value è invece una lista di rango 1 per

cui, secondo le regole sopra esposte, avverrà un'iterazione cartesiana su ciascuno dei suoi valori. Quindi per ciascun oggetto di Revit, la stessa proprietà verrà settata 4 volte, ogni volta sovrascrivendo il precedente valore che alla fine risulterà "Wa" (Figura 10).

Allo step 2 il nodo viene ripetuto 4 volte per ciascuna delle 4 ripetizioni dello step 1, quindi per un totale di 16 volte.

Utilizzando il nodo Element.GetParameterValueByName si può verificare il risultato finale inatteso (Figura 11), corrispondente alla situazione di partenza di Figura 4.

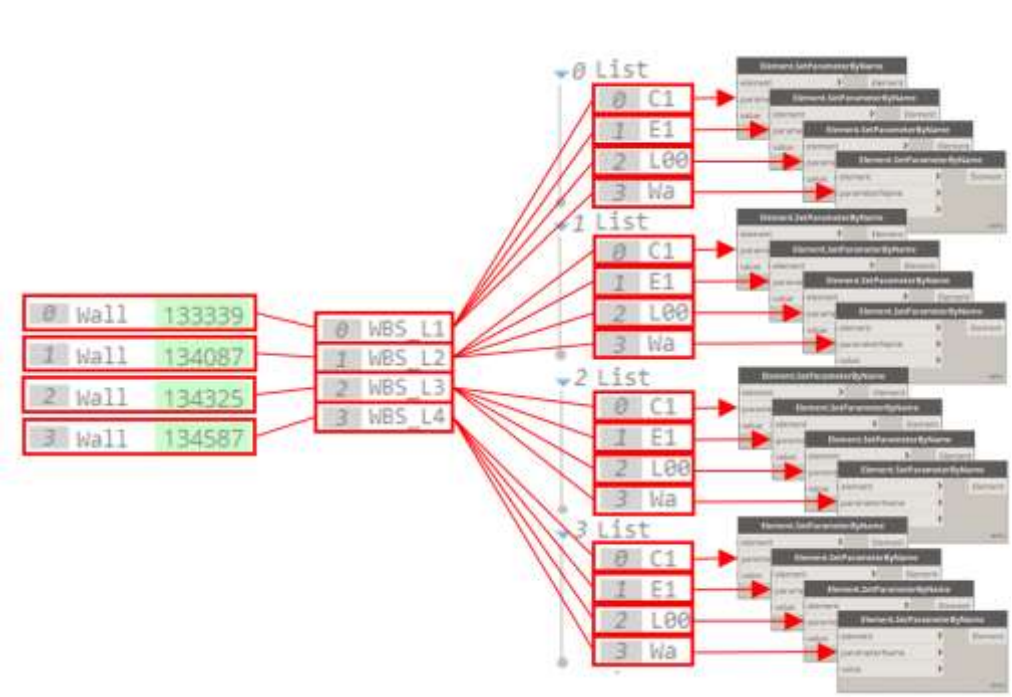

Figura 10

Figura 11

Conclusione

È evidente che la mancata comprensione delle regole di replicazione può portare a risultati inattesi e pericolosi. Nell'esempio esaminato il nodo non restituisce alcun avviso, perché di fatto le operazioni eseguite sono formalmente corrette, ma i valori impostati nelle proprietà degli oggetti di Revit sono errati ed incompleti. Il comportamento di default del nodo non è utile ai fini dell'esempio e in generale sorge l'esigenza di poter controllare le modalità con cui la replicazione viene eseguita. La riposta a questa esigenza sono le guide di replicazione che verranno trattate nel prossimo articolo.

Autore: Massimiliano Baraldo, BIM Manager & AEC Technical Specialist per One Team

Vuoi più informazioni sulla tematica?

Scrivici a mailto:marketing@oneteam.it