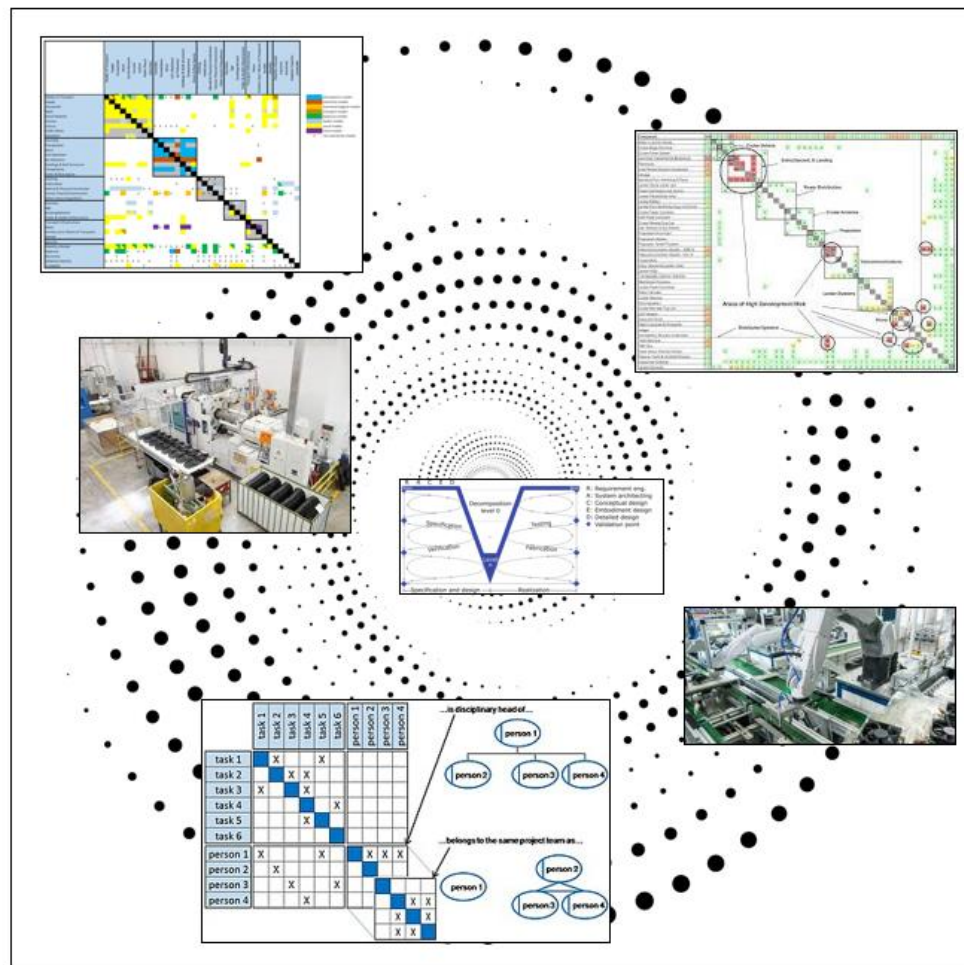


Breve introduzione alla Teoria DSM (Design Structure Matrix)



La DSM per la progettazione e l'efficientamento dei processi produttivi, dei flussi di materiali, degli scambi di informazione.

Materiale a cura del Dr. Daniele Grazzini – Versione Aprile 2024

INDICE

1 Introduzione e motivazioni:	4
2 Breve excursus storico:	5
3 Alcune definizioni matematiche fondamentali:	6
4 La teoria della DSM:	9
5 Leggere un DSM:	11
6 Tipologia o classificazione di DSM:	13
6.1 DSM basato su componenti:	14
6.2 DSM basato sul team:	16
6.3 DSM basato sulle attività:	17
6.4 DSM basato su parametri:	19
7 Approfondimento sul DSM basato sulle attività (o Schedule DSM):	20
7.1 Formazione del DSM:	20
7.1.1 Passaggi fondamentali suggeriti dai ricercatori del Mit di Boston:	21
7.2 Partizionamento di un DSM (Sequencing):	21
7.2.1 Identificare i loop in base al percorso di ricerca:	22
7.2.2 Metodo della potenza della matrice di adiacenza:	24
7.2.3 Il metodo della matrice di raggiungibilità:	25
7.3 “Tearing” ovvero “Lacerazione” di un DSM:	26
7.4 Simulazione del modello di iterazione:	26
7.4.1 Descrizione del modello proposto dal MIT di Boston e in particolare del lavoro di Browning ed Eppinger del 2001:	26
7.4.2 Sommario del modello e delle variabili di simulazione:	27
7.4.3 Algoritmo qualitativo del codice di simulazione:	27
7.4.4 Spiegazione dell’algoritmo di simulazione:	28
8 Software CAM Ver.2 (a cura dei ricercatori dell’Università di Cambridge)	30
8.1 Come costruire un modello DSM	30
8.2 Importazione di DSM da CSV (Excel)	31
8.3 Creazione di nuovi elementi (righe/colonne):	31
8.4 Analisi:	32
8.5 Risultati:	33
9 Fogli Excel con macro: DSM_Program-V2.1.xls (a cura del team di ricerca del MIT di Boston)	34
9.1 Introduzione a VBA:	34
9.2 Foglio “Element Info”:	34
9.3 Foglio “DSM”:	35
9.4 Foglio “Dependence Report”:	35
9.5 Foglio “Partitioned DSM”	36
9.6 Foglio “Banded DSM”:	36

9.7 Foglio “SIM Input”:	37
9.8 Foglio “Probability”:	37
9.9 Foglio “Impact”:	38
9.10 Foglio “SIM Results”:	38
9.11 Foglio “Single Run Data”:	38
9.12 Aggiornamento del file Excel con Macro del Mit di Boston:	39
9.13 Istruzioni di utilizzo della pagina “Activities-Parameters”:	42
9.14 Celle di riferimento importanti	45
9.15 Spiegazione dell’algoritmo per la formazione della matrice DSM:	46
10 Sviluppo del Plugin Obsidian “Adjacency Matrix Exporter”:	51
10.1 Il software Obsidian.md	52
10.2 Scopo e descrizione del funzionamento del plugin “Adjacency Matrix Exporter”	53
10.3 Spiegazione della struttura del plugin	55
10.3.1 Il metodo exportCsvAbsolute()	56
10.4 La funzionalità canvas di Obsidian.md	57
10.4.1 Aggiunta del titolo	57
10.4.2 Aggiunta di una annotazione	58
10.4.3 Aggiunta di una nota dal Vault	58
10.4.4 Aggiunta di un file multimediale dal Vault	58
10.4.5 Aggiunta di collegamenti:	59
10.4.6 Funzionalità di Zoom e scorrimento	59
10.4.7 Creazione di un gruppo	60
10.4.8 Esportazione della lavagna con un’immagine	60
10.4.9 Collegamento con il plugin “Adjacency Matrix Exporter”:	60
11 Bibliografia:	62
Ringraziamenti	64

1 Introduzione e motivazioni:

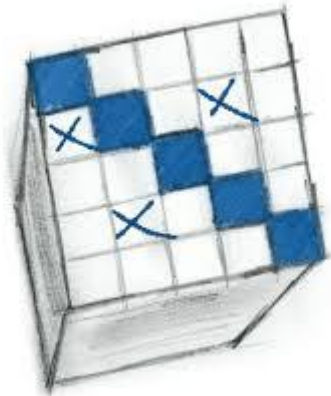
La dispensa proposta all'attenzione di coloro che assisteranno al seminario o che semplicemente si imbattono in questo lavoro (a volte la voglia e la sete di conoscenza ci porta in luoghi inaspettati...) riporta una breve panoramica della teoria generale del DSM (Design Structure Matrix) una metodica che rientra a pieno titolo nelle tecniche di Scheduling a disposizione di coloro che si occupano di project management e che affronta la tematica assai interessante dell'analisi dei sistemi complessi. L'interesse della comunità scientifica internazionale e di alcune tra le più prestigiose università del mondo tra cui il Mit di Boston nei confronti di questa materia, giustifica lo sforzo della composizione di queste pagine: ci auguriamo che gli spunti proposti e le possibilità che si apriranno da una lettura curiosa e attenta invoglierà all'approfondimento per poter ampliare gli attuali orizzonti della ricerca accademica. "Siamo consapevoli che tanto è stato fatto ma tantissimo rimane da fare e da scoprire", con queste parole i maggiori esperti del settore incoraggiano studiosi, accademici, appassionati e il mondo imprenditoriale produttivo a conoscere ed usare le tecniche proposte dalla DSM: la testimonianza maggiore in questo senso è dato dalle conferenze internazionali che ogni anno da ormai 25 anni si tengono in giro per il mondo e che raccolgono centinaia di paper e nuovi articoli scientifici.

Attualmente l'Università Telematica Interazionale Uninettuno, tramite alcuni lavori di tesi sperimentale proposti dal dipartimento di ingegneria, sta cercando di portare il proprio contributo sia nella direzione di un maggiore approfondimento della materia che nella creazione di sinergie e contatti all'interno di questa vasta comunità di studiosi e ricercatori. Ci auguriamo che lo stimolo proposto in queste pagine sia di ispirazione a studenti e studentesse che potranno esprimere il loro talento portando il proprio contributo originale.

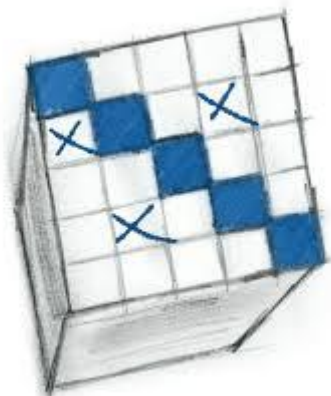
Il presente lavoro si articola in due parti principali: la parte prima si concentra sulla descrizione delle caratteristiche generali della teoria, si fa una classificazione in base alle applicazioni possibili, si approfondisce il DSM basato sulle attività entrando nel dettaglio del metodo. Dopo aver spiegato le 4 fasi applicative e il loro significato si tenta un avvicinamento al modello di simulazione delle iterazioni, implementato nei fogli Excel con macro proposto dai ricercatori del Mit di Boston in particolare dal Prof. Tyson R. Browning e dal Prof. Steven P. Eppinger; nella parte seconda si propone la conoscenza di tre strumenti informatici che si è avuto la possibilità di utilizzare nella composizione di lavori precedenti. In particolare, i primi due strumenti sono il frutto dell'elaborazione dei ricercatori del Mit di Boston: i fogli Excel con macro scaricabili dal seguente link <https://dsmweb.org/excel-macros-for-partitioning-and-simulation/> e il software CAM V.2.1 dell'Università di Cambridge scaricabile gratuitamente al seguente link <https://camtoolkit.eng.cam.ac.uk/>; Il terzo strumento è un mio contributo originale alla tematica della generazione di matrici DSM, un plugin sviluppato all'interno della piattaforma Obsidian.md utilizzando la programmazione Typescript entrato a far parte della collezione dei plugin scaricabili dalla Community di Obsidian attraverso un percorso di validazione conclusosi nel Dicembre 2023. Obsidian è scaricabile gratuitamente al link <https://obsidian.md/> e il plugin è scaricabile al link <https://obsidian.md/plugins> digitando nella barra di ricerca il nome "Adjacency Matrix Exporter" (autore Daniele Grazzini). Con questa carrellata di strumenti descritti in modo da facilitarne l'utilizzo si spera di avere dato un valido aiuto a coloro che si cimenteranno nello studio e nell'applicazione pratica delle competenze acquisite.

Auguro a tutti Voi una buona e proficua lettura!

2 Breve excursus storico:



Il termine DSM, coniato negli anni '70 dal professor Don Steward della California State University, Sacramento, rappresenta un fondamentale contributo allo sviluppo della metodologia DSM. Il formato a matrice quadrata è stato introdotto per rappresentare in modo efficace le interazioni tra variabili di progettazione. Nel 1989, al MIT di Boston, il DSM è stato riconosciuto per il suo potenziale. Negli anni '90 è stato dato l'avvio a progetti industriali nei settori automobilistico, elettronico e aerospaziale accumulando un ingente numero di casi studio e consulenze. L'esperienza acquisita ha portato all'estensione del DSM, includendo modelli architetturali statici, analisi di clustering e applicazioni diverse nei domini di prodotto e organizzazione. NASA, Boeing, General Motors e Intel sono state tra le prime a adottare e sviluppare il DSM in quegli anni. Attualmente, il DSM trova applicazioni diffuse in molteplici settori industriali. La comunità di ricerca DSM, fondata alla fine degli anni '90 attraverso workshop al MIT, si è estesa globalmente, coinvolgendo ricercatori da Europa, Asia, Australia, Sud America e Nord America. Questa comunità include anche sviluppatori di software, consulenti e utenti chiave nei settori in cui il DSM è attivamente impiegato. Numerosi documenti di ricerca, disponibili su www.dsmweb.org e www.dsm-conference.org, narrano lo sviluppo dei metodi DSM e documentano una vasta gamma di applicazioni.



3 Alcune definizioni matematiche fondamentali:

Entriamo nel vivo della materia attraverso questo paragrafo che fornisce le definizioni matematiche centrali per la comprensione della trattazione successiva. Si analizzano in particolare le nozioni di Teoria dei Grafi e di teoria delle Matrici con particolare riferimento a due argomenti: il digrafo e la matrice di adiacenza evidenziandone le caratteristiche peculiari e la loro relazione.

Un *grafo* è una coppia di insiemi $G=(V,A)$ in cui V è un insieme di v elementi definiti *vertici* o *nodi* ($V=\{v_1, v_2, v_3, \dots, v_v\}$) ed A è un insieme di a elementi definiti *archi* o *spigoli* o *edge* ($A=\{a_1, a_2, a_3, \dots, a_a\}$). Ciascun edge corrisponde ad una coppia di vertici (v_i, v_j) .

Si definisce *densità* di un grafo il rapporto tra il numero di edge ed il numero di vertici:

$d=|A|/|V|$ dove con il simbolo $| * |$ si è inteso scrivere la cardinalità dell'insieme relativo ossia la totalità dei suoi elementi.

Se la coppia di vertici (v_i, v_j) è ordinata, l'arco o edge si dice *orientato* dal vertice v_i origine al vertice v_j destinazione.

Se la coppia di vertici non è ordinata l'arco non è orientato e si indica anche con il termine *spigolo*.

Un grafo si dice *orientato* se l'insieme A è costituito da archi orientati. Si dice *non orientato* se l'insieme A è costituito da archi non orientati.

Nel seguito di questo capitolo si tratterà di grafi orientati.

Due vertici si dicono *adiacenti* se esiste un arco del grafo che li congiunge.

Due archi si dicono adiacenti se hanno una estremità in comune.

Un arco in cui l'origine coincide con la destinazione si definisce *cappio*.

Nella figura 3.1 viene mostrato un esempio di grafo orientato G .

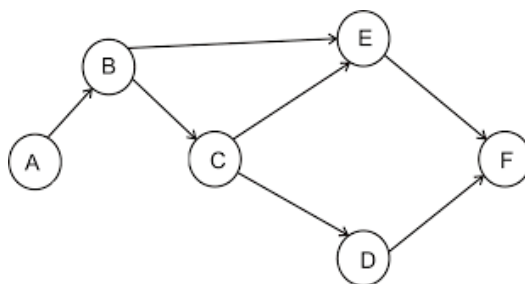


Fig. 3.1: esempio di grafo orientato

La forma di rappresentazione più immediata di un grafo è quella grafica, in cui i vertici o nodi sono rappresentati da punti del piano e gli archi da linee (orientate o non orientate). Essa è utile solo per una analisi visiva delle informazioni contenute in un grafo di piccole dimensioni. Quando il grafo è di dimensioni medio-grandi ed in ogni caso quando è necessaria una elaborazione numerica e algoritmica di queste informazioni, è opportuno adottare altre forme di rappresentazione utilizzabili per la elaborazione automatica dei dati.

Queste forme di rappresentazione sono basate sostanzialmente su una struttura dei dati di tipo matriciale.

In questo lavoro si farà uso della matrice di adiacenza.

La *matrice di adiacenza* A è una matrice binaria (o booleana 0/1) in cui ogni riga i è associata al vertice v_i assunto come origine ed ogni colonna j è associata al vertice v_j assunto come destinazione.

Ciascun termine a_{ij} assume i seguenti valori:

$a_{ij} = 0$ se l'arco $v_i - v_j$ non esiste, mentre è 1 se l'arco esiste

La matrice A descrive pertanto la struttura del grafo in termini di adiacenza dei nodi, cioè di assenza o presenza degli archi così come evidenziato in figura 3.2.

La somma degli elementi di una riga i fornisce il numero di archi che hanno origine nel vertice i .

La somma degli elementi di una colonna j fornisce il numero degli archi che hanno destinazione nel vertice j .

La presenza di elementi unitari sulla diagonale principale indica l'esistenza di archi che hanno origine e destinazione nello stesso vertice (cappi).

	A	B	C	D	E	F
A	0	1	0	1	0	0
B	0	0	1	0	0	0
C	0	0	0	0	1	0
D	1	0	1	0	0	0
E	0	0	0	1	0	0
F	0	0	0	0	0	0

Fig. 3.2: esempio di matrice binaria

Nel proseguo di questo lavoro si farà riferimento ad un grafo per il quale sono definite funzioni di tipo numerico sugli archi (vedi Fig. 3.2): è possibile, infatti, associare agli archi dei pesi corrispondenti a parametri di funzionamento del particolare collegamento tra vertici.

I pesi associati ad un arco ij esprimono in generale:

- una misura dello spostamento definito in generale costo c_{ij} dello spostamento per unità di flusso, se C è l'insieme dei costi di spostamento sugli archi della rete;
- ai parametri di capacità dell'arco, in termini fisici (sezione stradale, diametro di un tubo o di un cavo) oppure in termini di flusso massimo sull'arco (veicoli/h, m^3/s , Mbit/s);
- ad una misura del costo di costruzione e/o manutenzione e gestione dell'arco, indicato con h_{ij}

Di particolare interesse sarà quindi la matrice di adiacenza del grafo nel quale ciascun elemento a_{ij} assume dei valori ossia dei pesi.

Applichiamo quanto finora visto ad un esempio di attività di processo riprendendo man mano le definizioni importanti viste.

Consideriamo un sistema composto da tre attività: attività “A”, attività “B” ed attività “C” e dalle relazioni tra queste attività rappresentate da frecce con verso. La relazione potrebbe essere che l’elemento di arrivo “dipende da” l’elemento di partenza.

È possibile sviluppare un grafico per rappresentare pittoricamente questo sistema come mostrato in figura 3.3 (a). Il grafo risultante è chiamato grafo diretto o semplicemente digrafo.

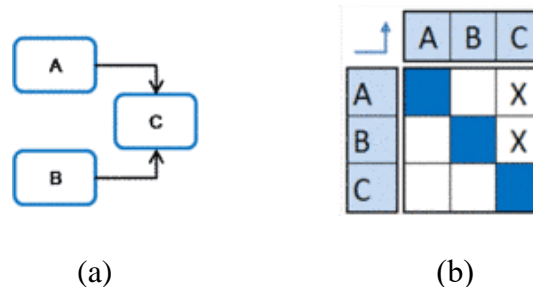


Fig. 3.3: (a) esempio di digrafo, (b) relativa matrice di adiacenza

La rappresentazione matriciale di un digrafo (cioè un grafo orientato così come si evince dalla figura 3.3 (b)) viene definita matrice di adiacenza ed ha le seguenti proprietà:

- è binaria (cioè, una matrice popolata solo con zeri e uno)
- è quadrata (cioè, una matrice con lo stesso numero di righe e colonne)
- ha n righe e colonne (n è il numero di nodi del digramma nel nostro caso delle attività)
- ha k elementi diversi da zero, (k è il numero di archi nel digrafo)

Il layout della matrice è il seguente: i nomi degli elementi del sistema sono posizionati lungo il lato della matrice come intestazioni di riga e nella parte superiore come intestazioni di colonna nello stesso ordine.

Se esiste un collegamento dal nodo i al nodo j , allora il valore dell'elemento ij (riga i , colonna j) è unitario (o contrassegnato con una X).

Altrimenti, il valore dell'elemento è zero (o lasciato vuoto).

Nella rappresentazione a matrice binaria di un sistema, gli elementi diagonali della matrice non hanno alcuna interpretazione nel descrivere il sistema, quindi di solito vengono lasciati vuoti o evidenziati con un colore.

Le matrici binarie sono utili nella modellazione dei sistemi perché possono rappresentare la presenza o l'assenza di una relazione tra coppie di elementi in un sistema. Uno dei principali vantaggi della rappresentazione a matrice rispetto al digrafo è che fornisce una informazione compatta e una mappatura sistematica tra gli elementi del sistema consentendo un'analisi dettagliata di un insieme limitato di elementi nel contesto della struttura complessiva.

Se il sistema è un progetto rappresentato da un insieme di attività da eseguire, come nel nostro caso specifico, allora i segni fuori diagonale in una singola colonna del DSM rappresentano tutte le attività il cui output è richiesto per eseguire l'attività corrispondente a quella colonna (ad esempio, leggere una colonna per visualizzare gli input di un'attività, che sono gli output di altre attività).

Allo stesso modo, leggere lungo una riga specifica rivela quali attività ricevono informazioni dall'attività corrispondente a quella riga (ad esempio, leggere lungo una riga per vedere dove vanno gli output di un'attività per diventare input di altre attività). In molti casi, e questo sarà il nostro, l'ordine delle attività lungo la matrice corrisponde a una sequenza temporale.

4 La teoria della DSM:



La Dependency Structure Matrix o Design Structure Matrix (DSM) è un metodo versatile per rappresentare e analizzare modelli di sistema attraverso una matrice quadrata che evidenzia le relazioni tra gli elementi del sistema.

A differenza di altri metodi di modellazione, la DSM offre due vantaggi chiave:

1. Semplice e Conciso: fornisce una rappresentazione chiara di sistemi complessi.
2. Analisi Potenti: permette analisi avanzate come il clustering per favorire la modularità e il sequenziamento per minimizzare costi e pianificare il rischio.

Ad esempio, considerando un sistema con elementi A, B, e C, le relazioni tra questi sono rappresentate in un grafo diretto o digrafo. La rappresentazione matriciale del digrafo è binaria, quadrata, e presenta k elementi diversi da zero, dove k è il numero di archi nel digrafo. Il layout della matrice associa i nomi degli elementi alle righe e colonne, con valori unitari se esiste una connessione tra i nodi. Questo approccio binario offre un'informazione compatta, consentendo un'analisi dettagliata di elementi specifici all'interno della struttura complessiva. Le matrici binarie sono efficaci per modellare la presenza o assenza di relazioni tra elementi. Tuttavia, è possibile estendere questa rappresentazione utilizzando una DSM numerica per indicare dipendenze ponderate, aggiungendo una colonna per rappresentare il peso di un elemento. Questa flessibilità consente una modellazione più dettagliata delle interazioni nel sistema. Bisogna tenere presente che i DSM non sono scritti tutti in modo univoco; esiste infatti una convenzione duale secondo cui gli output e gli input sono scambiati: si tratta della trasposizione della matrice della prima convenzione. Entrambe le convenzioni trasmettono informazioni equivalenti e chiunque abbia familiarità con una convenzione può solitamente adattarsi all'altra abbastanza rapidamente e facilmente.

Scendiamo ora nel dettaglio della comprensione della matrice DSM.

Esistono tre modi per descrivere le relazioni tra gli elementi (attività nel nostro caso) della matrice come mostrato nella figura 4.1: parallele (o simultanee), sequenziali (o dipendenti) e accoppiate (o interdipendenti).

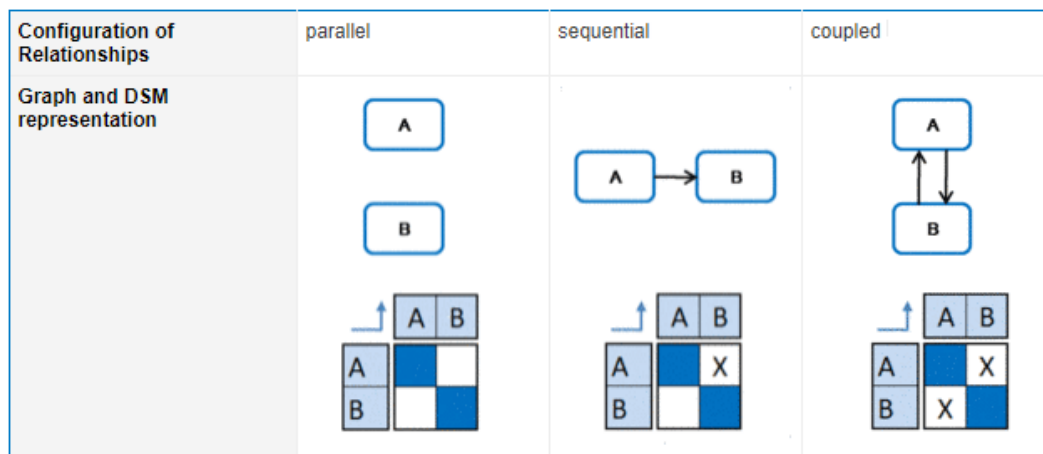


Fig.4.1: schema delle possibili relazioni tra gli elementi della matrice

Nella configurazione parallela gli elementi del sistema non interagiscono tra loro (almeno non per il tipo di rappresentazione che viene rappresentata nel digrafo). Comprendere il comportamento dei singoli elementi ci permette di comprendere meglio il comportamento del sistema: nella configurazione parallela il compito B è indipendente dal compito A e non è richiesto alcuno scambio di informazioni tra le due attività; nella configurazione sequenziale, un elemento influenza il comportamento o la decisione di un altro elemento in modo unidirezionale. Ad esempio, i parametri di progettazione dell'elemento B vengono selezionati in base ai parametri di progettazione dell'elemento A; Infine, nel sistema accoppiato, il flusso di influenza o informazione è intrecciato in quanto l'elemento A influenza B e l'elemento B influenza A. Questa dipendenza ciclica è chiamata "Circuito" o "Ciclo". Uno degli aspetti più potenti dell'analisi DSM rispetto alle altre tecniche di Scheduling, come ad esempio il CPM (Critical Path Method, visto in corsi precedenti), è la possibilità di gestire i cicli attraverso tecniche che approfondiremo più avanti.

5 Leggere un DSM:



Uno dei vantaggi principali del DSM è la natura grafica del formato di visualizzazione a matrice. La matrice fornisce una rappresentazione altamente compatta, facilmente scalabile e leggibile in modo intuitivo di un'architettura di sistema. La figura 5.1 (a) mostra un semplice modello DSM di un sistema con sei elementi. Le celle lungo la diagonale della matrice rappresentano gli elementi del sistema. Per mantenere compatto il diagramma a matrice, i nomi completi degli elementi sono spesso elencati a sinistra delle righe (e talvolta anche sopra nelle colonne) anziché nelle celle diagonali. È anche facile pensare che ciascuna cella diagonale abbia potenzialmente input che entrano dalla sua parte superiore e inferiore e output che escono dai suoi lati sinistro e destro. Le fonti e le destinazioni di queste interazioni di input e output sono identificate da segni nelle celle fuori diagonale. L'esame di qualsiasi riga nella matrice rivela tutti gli output dell'elemento in quella riga (che sono input per altri elementi). Guardando verso il basso qualsiasi colonna della matrice vengono mostrati tutti gli input per l'elemento in quella colonna (che sono output da altri elementi). Ad esempio, nella figura 5.1 (a), leggendo la riga 2, vediamo che l'elemento 2 fornisce output agli elementi 3 e 4. Leggendo la colonna 5, vediamo che l'elemento 5 riceve input dagli elementi 1, 3 e 4. Pertanto, a Il segno in una cella fuori diagonale (ad esempio, cella 3,5) rappresenta un'interazione che è sia un input che un output, a seconda che si assuma la prospettiva del fornitore (elemento 3) o del destinatario (elemento 5).

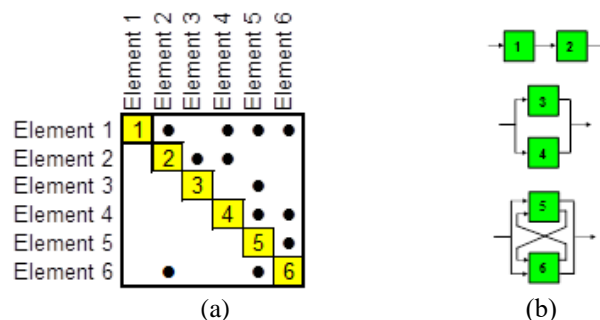


Fig. 5.1: (a) matrice DSM, (b) relativa interpretazione grafica delle relazioni della matrice

A destra del DSM sopra riportato vedi figura 5.1 (b) ci sono gli equivalenti del diagramma di collegamento diretto ai nodi di parti del DSM. Si noti che gli elementi 1 e 2 formano una catena o sequenza lineare, mentre gli elementi 3 e 4 sono indipendenti e gli elementi 5 e 6 sono interdipendenti

o accoppiati. Questo semplice esempio di DSM è chiamato DSM binario perché i segni fuori diagonale indicano semplicemente la presenza o l'assenza di un'interazione. La rappresentazione binaria del DSM può essere estesa in molti modi, così come detto in precedenza, includendo ulteriori attributi delle interazioni, come il numero di interazioni e/o l'importanza, l'impatto o la forza di ciascuna, che potrebbero essere rappresentati utilizzando uno o più valori numerici, simboli, sfumature o colori invece dei soli segni binari in ciascuna delle celle fuori diagonale. Questa forma estesa di DSM è chiamata DSM numerico. Ulteriori attributi degli elementi stessi possono essere inclusi anche aggiungendo più colonne a sinistra della matrice quadrata per descrivere, ad esempio, il tipo o lo stato di ciascun elemento.

6 Tipologia o classificazione di DSM:



Sono stati identificati quattro tipi principali di dati che possono essere rappresentati in un DSM, sebbene sia possibile adattare la rappresentazione alle proprie esigenze di studio come mostra la tabella 6.1. Quelli elencati costituiscono i tipi principali di DSM che sono stati studiati, come detto in precedenza, attraverso numerosi casi studio. Vedremo nei paragrafi successivi di dire qualcosa di più su ciascuno di questi tipi cercando di metterne in luce le caratteristiche principali.

Tipi di dati DSM	Rappresentazione	Applicazioni
Basato su componenti	Relazioni tra componenti	Architettura, ingegneria e progettazione di sistemi
Basato sulle persone	Relazioni tra unità organizzative	Progettazione organizzativa, gestione delle interfacce, integrazione del team
Basato sulle attività	Relazioni input/output delle attività	Miglioramento dei processi, pianificazione dei progetti, gestione dell'iterazione, gestione del flusso di informazioni
Basato su parametri	Relazioni tra i parametri di progettazione	Sequenziamento delle attività di basso livello e costruzione del processo, sequenziamento delle decisioni di progettazione

Tabella 6.1: classificazione DSM

6.1 DSM basato su componenti:

Un DSM basato su componenti documenta le interazioni tra gli elementi in un'architettura di sistema complessa. Nel DSM possono essere visualizzati diversi tipi di interazioni. I tipi di interazioni varieranno da progetto a progetto. Alcuni tipi di interazione rappresentativi sono mostrati nella tabella 6.1.1:

Spaziale	esigenze di adiacenza o orientamento tra due elementi
Energia	esigenze di trasferimento/scambio di energia tra due elementi
Informazione	esigenze di scambio di dati o segnali tra due elementi
Materiale	esigenze di scambio materiale tra due elementi

Tabella 6.1.1: tipi di interazioni nel DSM basato su componenti

Sono possibili anche altre classificazioni. Un altro elenco completo delle dipendenze di modellazione in un'architettura di prodotto è fornito dalla tabella 6.1.2:

Stato stazionario meccanico	I componenti sono in contatto fisico e impongono l'uno sull'altro un carico meccanico stabile. Questa è una relazione simmetrica.
Dinamica meccanica	I componenti sono in contatto e interagiscono attraverso una forza fluttuante o uno spostamento. Questa può essere una relazione direzionale.
Spaziale	I componenti si toccano oppure l'adiacenza e l'orientamento sono importanti. Questa è una relazione simmetrica.
Stato stazionario termico	Esiste una differenza di temperatura di stato stazionario tra i due componenti. Questa può essere una relazione direzionale.
Dinamica termica	C'è una differenza di temperatura fluttuante tra i due componenti. Questa può essere una relazione direzionale.
Segnale elettrico	Un segnale passa da un componente all'altro. Questa può essere una relazione direzionale.
Terra elettrica	Esiste un collegamento elettrico a terra tra i due componenti. Questa può essere una relazione direzionale.
Dinamica elettrica	La progettazione fisica o il comportamento logico di un componente è collegato alla progettazione fisica o al comportamento logico dell'altro. Questa può essere una relazione direzionale.

Tabella 6.1.2: elenco delle dipendenze di modellazione in un DSM basato su componenti

Ad esempio, consideriamo l'interazione materiale tra i componenti di un sistema di climatizzazione di un'automobile. In questo caso, la ventola del motore (B) deve trasferire materiale al condensatore (E), poiché è presente una "X" nella cella (B, E). La situazione è visibile nella figura 6.1.3.

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator	A	X															
Engine Fan	B		X			X											
Heater Core	C			X													X
Heater Hoses	D				X												
Condenser	E					X	X		X								
Compressor	F						X	X		X	X						
Evaporator Case	G							X									X
Evaporator Core	H					X	X		X								X
Accumulator	I						X	X		X							
Refrigeration Controls	J									X							
Air Controls	K										X						
Sensors	L											X					
Command Distribution	M												X				
Actuators	N													X			
Blower Controls	O														X		
Blower Monitor	P				X			X	X							X	

Fig 6.1.3: esempio di matrice DSM basato su componenti

La matrice può ora essere riorganizzata per ottenere cluster di componenti altamente interagenti tentando di ridurre al minimo le interazioni tra cluster così come mostrato in figura 6.1.5. In questo modo i dati non vengono modificati, ma le righe e le colonne della matrice vengono solo scambiate a coppie per ottenere un diverso layout della matrice (di questo metodo si parlerà in seguito). I raggruppamenti ottenuti rappresentano un quadro utile per riorganizzare l'architettura del prodotto e focalizzare l'attenzione sulle interfacce tra i moduli.

Il raggruppamento dei segni "X" lungo la diagonale del DSM ha portato alla creazione di tre cluster per il sistema di controllo climatico. Questi cluster rappresentano gruppi di componenti strettamente interconnessi e sono visibili per maggiore chiarezza in tabella 6.1.4. Possono essere utilizzati per definire moduli che possono, ad esempio, essere ordinati da diversi fornitori di sistemi o che possono essere utilizzati su una serie di frigoriferi diversi (di volume piccolo, medio, grande, ad esempio) come moduli di riporto con interfacce ben definite per gli altri moduli (=cluster).

Gruppo 1	Parte anteriore dell'aria
Gruppo 2	Pezzo refrigerante
Gruppo 3	Blocco d'aria interno

Tabella 6.1.4: descrizione dei cluster individuati

		D	J	K	L	M	N	A	B	E	F	I	H	C	P	O	G
Heater Hoses	D	X															
Refrigeration Controls	J		X														
Air Controls	K			X													
Sensors	L				X												
Command Distribution	M					X											
Actuators	N						X										
Radiator	A							X	X								
Engine Fan	B								X	X							
Condenser	E									X	X	X	X				
Compressor	F										X	X	X				
Accumulator	I											X	X				
Evaporator Core	H									X	X	X		X			
Heater Core	C														X		
Blower Monitor	P													X	X	X	X
Blower Controls	O														X	X	
Evaporator Case	G															X	

Fig. 6.1.5: matrice DSM con i Cluster evidenziati

6.2 DSM basato sul team:



Questo approccio viene utilizzato per l'analisi e la progettazione organizzativa basata sul flusso di informazioni tra varie entità organizzative. Gli individui e i gruppi che partecipano ad un progetto sono gli elementi analizzati (righe e colonne nella matrice). Un DSM basato su team viene costruito identificando i flussi di comunicazione richiesti e rappresentandoli come connessioni tra entità organizzative nella matrice. Per l'esercizio di modellazione è importante specificare cosa si intende per flusso di informazioni tra i team. La tabella 6.2.1 presenta diversi modi possibili in cui è possibile caratterizzare il flusso di informazioni.

Tipo di flusso	Possibili metriche
Livello di dettaglio	Da scarsi (documenti, e-mail) a ricchi (modelli, incontri faccia a faccia)
Frequenza	Da basso (batch, puntuale) ad alto (online, reale)
Direzione	Da sola andata a bidirezionale
Tempistica	Da precoce (preliminare, incompleto, parziale) a tardivo (finale)

Tabella 6.2.1: possibili modi per caratterizzare il flusso di informazioni

Ancora una volta, la matrice può essere manipolata per ottenere cluster di team e individui altamente interagenti tentando di ridurre al minimo le interazioni tra cluster. I raggruppamenti ottenuti rappresentano un quadro utile per la progettazione organizzativa concentrandosi sulle esigenze comunicative previste dei diversi attori.

6.3 DSM basato sulle attività:



Consideriamo una serie di attività in un processo e prendiamo ad esempio la situazione descritta in figura 6.3.1. Questi compiti devono lavorare insieme per raggiungere l'obiettivo del processo complessivo. Lo scambio di informazioni può quindi essere rappresentato come un digramma o un DSM.

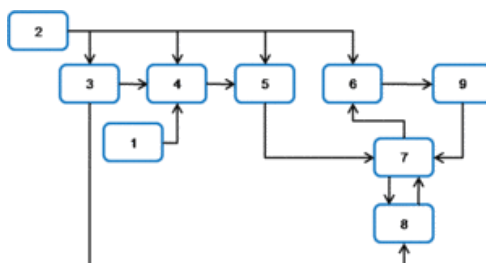


Fig. 6.3.1: esempio di relazioni tra attività di processo

Dalla matrice (si invita lo studente a verificare il passaggio dal grafo alla matrice) si possono osservare tre tipi di interazioni tra compiti. Nella figura 6.3.2, i compiti 1 e 2 sono “indipendenti” poiché non viene scambiata alcuna informazione tra loro, lo stesso vale per gli elementi 4 e 8. Questi compiti possono essere eseguiti ciascuno simultaneamente (in parallelo). I compiti 3, 4 e 5 sono impegnati in un trasferimento sequenziale di informazioni e sono considerati “dipendenti”. Queste attività verrebbero generalmente eseguite in serie. I compiti 7 e 8, tuttavia, dipendono reciprocamente dalle informazioni. Si tratta di attività “interdipendenti” o “accoppiate” che spesso richiedono più iterazioni per il completamento. In definitiva, i compiti 6, 7 e 9 sono impegnati in un ciclo.

	1	2	3	4	5	6	7	8	9
1				X					
2			X	X	X	X			
3				X				X	
4					X				
5						X			
6							X		X
7						X		X	
8							X		
9							X		

Fig. 6.3.2: matrice DSM basata sulle attività

Le celle contrassegnate sotto la diagonale rappresentano potenziali cicli di rilavorazione o iterazioni nel processo. Ciò si verifica quando un'attività dipende dalle informazioni di un'attività pianificata per un'esecuzione successiva. Tali scenari spesso portano a rielaborazioni e sono indesiderabili. Sono stati sviluppati numerosi algoritmi per ridurre al minimo tali casi di iterazione (celle contrassegnate sub-diagonali) riorganizzando la sequenza delle attività nel processo. Sono inoltre disponibili metodi per gestire le iterazioni del processo che non possono essere eliminate mediante la riseenquenziamento. Comunemente, gli algoritmi di sequenziamento di base vengono definiti “triangolarizzazione”, poiché l'obiettivo è ottenere una “matrice triangolare superiore” che preferibilmente non abbia segni sotto la diagonale. I modelli DSM che utilizzano semplici rappresentazioni binarie mostrano l'esistenza di una dipendenza tra due attività senza fornire informazioni aggiuntive sulla natura dell'interazione. Ulteriori studi hanno ampliato la configurazione di base del DSM acquisendo fatti aggiuntivi sul processo di sviluppo. Ad esempio, il DSM numerico sostituisce i segni con numeri nelle celle fuori diagonale per rappresentare il grado di dipendenza tra due compiti. Ciò rende possibile mostrare, ad esempio, la probabilità di un ciclo di feedback e quindi dare priorità a importanti iterazioni nella pianificazione del processo. Ritourneremo con un approfondimento successivo su questo particolare caso di DSM.

6.4 DSM basato su parametri:



Questo tipo di modellazione viene utilizzata per analizzare un processo di progettazione a livello delle relazioni tra i parametri. Nell'esempio sotto riportato si ha l'applicazione di un DSM basato su parametri alla progettazione di un sistema frenante automobilistico, utilizzando il DSM per descrivere le pratiche attuali di un fornitore di componenti di un sistema frenante. Il DSM di figura 6.4.1 è estratto dal DSM originale che era (103 per 103). Dopo aver sequenziato i parametri, nel DSM risultante diventano evidenti due blocchi di parametri (= cluster) così come si evince da figura 6.4.2.

		1	2	3	4	5	6	7	8	9	10	11	12	13
Customer Requirements	1			X	X	X								
Wheel Torque	2						X	X		X				X
Pedal Mech. Advantage	3					X		X		X		X		X
System Level Parameters	4		X	X		X		X	X	X	X	X		X
Rotor Diameter	5			X				X				X		X
ABS Modular Display	6													
Front Lining Coef. of Friction	7					X								X
Piston-Rear Size	8			X		X		X			X	X		X
Caliper Compliance	9						X							
Piston- Front Size	10			X		X		X	X	X		X		X
Rear Lining Coef of Friction	11					X							X	
Booster - Max. Stroke	12													
Booster Reaction Ratio	13			X		X		X		X		X	X	

Fig. 6.4.1: DSM basato su parametri originario

		1	4	2	10	8	3	11	7	13	5	12	9	6
Customer Requirements	1		X				X				X			
System Level Parameters	4			X	X	X	X	X	X	X	X		X	
Wheel Torque	2				X	X				X	X			X
Piston- Front Size	10					X	X	X	X	X	X		X	
Piston-Rear Size	8				X		X	X	X	X	X			
Pedal Mech. Advantage	3						X	X	X	X			X	
Rear Lining Coef. of Friction	11							X	X		X			
Front Lining Coef. of Friction	7								X	X				
Booster Reaction Ratio	13					X	X	X		X	X	X		
Rotor Diameter	5						X	X	X	X				
Booster - Max. Stroke	12													
Caliper Compliance	9													X
ABS Modular Display	6													

Fig. 6.4.2: DSM basato su parametri dopo il sequenziamento

7 Approfondimento sul DSM basato sulle attività (o Schedule DSM):

Questo paragrafo ci consente un approfondimento metodologico del DSM basato sulle attività. Partiamo dicendo che un utilizzo consapevole del DSM basato sulle attività presenta numerosi vantaggi e può fornire informazioni in diverse aree. Fornisce innanzitutto la visibilità del processo attraverso un rapido esame delle righe e delle colonne: facilita la gestione delle interfacce tra le attività e aiuta a tenere traccia delle modifiche che si vuole apportare. Evidenzia i feedback e le potenziali iterazioni che questi possono causare. Aiuta a evidenziare attività o aspetti non essenziali del flusso produttivo: costruire un processo di progettazione attorno al flusso di dati necessari piuttosto che tra le attività tradizionali può eliminare attività senza valore aggiunto e porzioni di esse supportando lo snellimento dei processi stessi. La trattazione che segue illustra le fasi che è bene seguire per arrivare a comporre una buona analisi. Si articola in 4 fasi successive che vengono analizzate nel dettaglio di seguito:

1. Formazione del DSM
2. Partizionamento del DSM
3. Strappo o lacerazione (Tearing)
4. Simulazione del modello di iterazione

7.1 Formazione del DSM:

Il successo del metodo DSM è determinato da un'adeguata scomposizione del sistema e dall'accuratezza delle relazioni di dipendenza. Pertanto, è fondamentale scomporre attentamente il sistema in esame in un insieme completo di elementi significativi del sistema. Una scomposizione appropriata può essere stabilita riunendo un gruppo di manager/esperti provenienti da diversi gruppi funzionali di un'organizzazione e chiedendo loro di elencare collettivamente i diversi sottosistemi che compongono il sistema nel suo complesso. La scomposizione può essere gerarchica o non gerarchica (a volte chiamata decomposizione di rete). Nella scomposizione gerarchica il sistema può essere suddiviso in sottosistemi o moduli. Tali moduli sono a loro volta suddivisi in componenti più fini, è quindi opportuno che la scomposizione sia netta e non ambigua. Nella scomposizione della rete, una gerarchia di sistema non è evidente, tale scomposizione è utile se non è possibile ideare una gerarchia semplice o se la gerarchia è ambigua. Una volta identificati gli elementi di sistema appropriati o l'insieme di attività che compongono un progetto, questi vengono elencati nel DSM come etichette di righe e colonne nello stesso ordine. Gli elementi all'interno della matrice vengono poi identificati chiedendo al manager o all'esperto appropriato del gruppo l'insieme minimo di parametri che influenzano il proprio sottosistema e contribuiscono al suo comportamento. In un DSM basato sulle attività, questo può essere l'insieme minimo di attività che devono essere eseguite prima che l'attività oggetto dell'interrogatorio possa essere avviata. In un DSM basato su parametri, le righe e le colonne sono parametri di progettazione che guidano la progettazione o definiscono il sistema e ai manager/esperti può quindi essere chiesto di definire le relazioni di precedenza tra i parametri elencati. Queste attività/parametri/elementi sono contrassegnate nel DSM da una "X".

7.1.1 Passaggi fondamentali suggeriti dai ricercatori del Mit di Boston:

1. Intervistare ingegneri e manager;
2. Verificare la presenza di possibili origini dati che possono essere analizzate o esportate in un DSM;
3. Determinare l'elenco degli elementi del sistema;
4. Chiedere informazioni sugli input, sugli output, sui punti di forza dell'interazione, ecc. tra gli elementi;
5. Inserire i dati nella matrice;
6. Raccogliere i commenti che spiegano ogni elemento e ogni dipendenza (per la successiva comprensione e interpretazione);
7. Consultare ingegneri e manager per verificare/commentare DSM;
8. Affinare il modello nel tempo assimilando l'apprendimento organizzativo.

7.2 Partizionamento di un DSM (Sequencing):

Il sequenziamento è il riordino delle righe e delle colonne del DSM in modo tale che la nuova disposizione del DSM non contenga alcun segno di feedback, trasformando così il DSM in una forma triangolare superiore. Per i sistemi ingegneristici complessi, è altamente improbabile che la semplice manipolazione di righe e colonne dia luogo ad una forma triangolare superiore. L'obiettivo dell'analista cambia quindi dall'eliminare i segni di feedback allo avvicinarli il più possibile alla diagonale (questa forma della matrice è detta triangolare a blocchi). Allo stesso modo, è possibile conoscere quali elementi del sistema potrebbero eventualmente dover essere rielaborati (ad esempio suddivisi in due elementi o eventualmente rimossi) per ottenere una migliore architettura di processo. Esistono diversi approcci utilizzati nel sequenziamento DSM. Tuttavia, sono tutti simili con una differenza nel modo in cui identificano i cicli (loop o circuiti) di elementi accoppiati. Tutti gli algoritmi di sequenziamento procedono come segue:

1. Identificare gli elementi del sistema (o attività) che possono essere determinati (o eseguiti) senza input dal resto degli elementi nella matrice. Questi elementi possono essere facilmente identificati osservando una colonna vuota nel DSM. Posiziona questi elementi a sinistra del DSM. Una volta riorganizzato un elemento, viene rimosso dal DSM (con tutti i segni corrispondenti) e il passaggio 1 viene ripetuto sugli elementi rimanenti.

2. Identificare gli elementi del sistema (o attività) che non forniscono informazioni ad altri elementi nella matrice. Tali elementi possono essere facilmente identificati osservando una riga vuota nel DSM. Posiziona questi elementi a destra del DSM. Una volta riorganizzato un elemento, viene rimosso dal DSM (con tutti i contrassegni corrispondenti) e il passaggio 2 viene ripetuto sugli elementi rimanenti.

3. Se dopo i passaggi 1 e 2 non rimangono elementi nel DSM, allora la matrice è completamente partizionata; altrimenti i restanti elementi contengono dei loop o cicli (almeno uno).

4. Determinare i cicli con uno dei seguenti metodi:

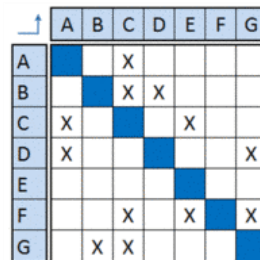
- Ricerca del percorso
- Metodo della potenza della matrice di adiacenza
- Il metodo della matrice di raggiungibilità
- Algoritmo di triangolarizzazione

5. Comprimere gli elementi coinvolti in un singolo loop in un elemento rappresentativo e tornare al passaggio 1.

7.2.1 Identificare i loop in base al percorso di ricerca:

Nella ricerca del percorso, il flusso di informazioni viene tracciato all'indietro o in avanti fino a quando un'attività non viene incontrata due volte. Tutte le attività tra la prima e la seconda occorrenza dell'attività costituiscono un ciclo di flusso di informazioni. Quando tutti i cicli sono stati identificati e tutte le attività sono state pianificate, la sequenza è completa e la matrice è in forma triangolare a blocchi. La figura 7.2.1.1 è un semplice esempio. La partizione di ricerca del percorso procede come segue:

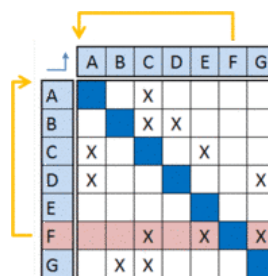
1. La matrice non partizionata è nel suo ordine originale.



	A	B	C	D	E	F	G
A			X				
B			X	X			
C	X				X		
D	X						X
E							
F			X		X		X
G		X	X				

Fig. 7.2.1.1: matrice DSM

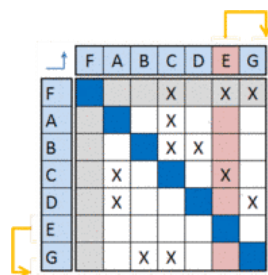
2. L'attività F non dipende dalle informazioni di altre attività, come indicato da una colonna vuota. Pianificare prima l'attività F nella matrice così come evidenziato in figura 7.2.1.2 e rimuoverla da ulteriori considerazioni.



	A	B	C	D	E	F	G
A			X				
B			X	X			
C	X				X		
D	X						X
E							
F			X		X		X
G		X	X				

Fig. 7.2.1.2: spostamento colonna e riga F

3. L'attività E, come si vede in figura 7.2.1.3, non fornisce informazioni ad alcuna attività nella matrice come indicato da una riga vuota. Pianificare l'attività E per ultima nella matrice e rimuoverla da ulteriori considerazioni.



	F	A	B	C	D	E	G
F				X		X	X
A				X			
B				X	X		
C	X					X	
D	X						X
E							
G			X	X			

Fig. 7.2.1.3: spostamento colonna e riga E

4. Ora nessuna attività ha righe o colonne vuote. Esiste un ciclo che può essere tracciato a partire da una qualsiasi delle attività rimanenti. In questo caso, selezioniamo l'attività A (arbitraria) e tracciamo la sua dipendenza dall'attività C. L'attività C dipende contemporaneamente dalle informazioni dell'attività A. Poiché l'attività A e l'attività C sono in un ciclo, comprimili l'una nell'altra e

rappresentare in un unico compito composito (ossia il compito CA).

	F	A	B	C	D	G	E
F					X		X X
A				X			
B				X X			
C		X					X
D		X				X	
G			X X				
E							

Fig. 7.2.1.4: dipendenza tra le attività A e C

5. L'attività CA, come evidenziato in figura 7.2.1.5, ha una colonna vuota che indica che non fa parte di nessun altro ciclo. Pianificalo per ultimo e rimuovilo da ulteriori considerazioni.

	F	CA	B	D	G	E
F		X			X X	
CA	X					X
B		X		X		
D		X			X	
G		X X				
E						

Fig. 7.2.1.5: attività CA

6. Traccia la dipendenza a partire da qualsiasi attività non pianificata: l'attività B dipende dall'attività G che dipende dall'attività D che dipende dall'attività B. Questo ciclo finale include tutte le restanti attività non pianificate.

	F	B	D	G	CA	E
F				X X X		
B			X		X	
D				X X		
G		X			X	
CA						X
E						

Fig. 7.2.1.6: evidenziazione del ciclo B-D-G

7. La matrice partizionata finale, vedi figura 7.2.1.7.

	F	B	D	G	C	A	E
F				X X			X
B			X		X		
D				X		X	
G		X			X		
C						X X	
A					X		
E							

Fig. 7.2.1.7: matrice DSM finale

7.2.2 Metodo della potenza della matrice di adiacenza:

La matrice di adiacenza è un DSM binario, vedi figura 7.2.2.1 (a), in cui una cella vuota viene sostituita con uno "zero" e una cella non vuota viene sostituita con "uno". Elevando il DSM alla n-esima potenza mostra quale elemento può essere raggiunto da se stesso in n passi osservando una voce diversa da zero per quel compito lungo la diagonale della matrice. Ad esempio, il quadrato del DSM (come si osserva in figura 7.2.2.1 (b)) mostra che le attività A e C sono coinvolte in un ciclo in due fasi. Si noti che nella matrice quadrata risultante, le celle con un valore maggiore di uno sono state sostituite da un valore pari a uno. Allo stesso modo, cubando il DSM, come mostrato in figura 7.2.2.1 (c), mostra che i compiti B, D ed E sono coinvolti in un ciclo di tre fasi. Le potenze più elevate del DSM non rivelano altri anelli nel sistema, così come si vede con la figura 7.2.2.1 (d) e 7.2.2.1 (e).

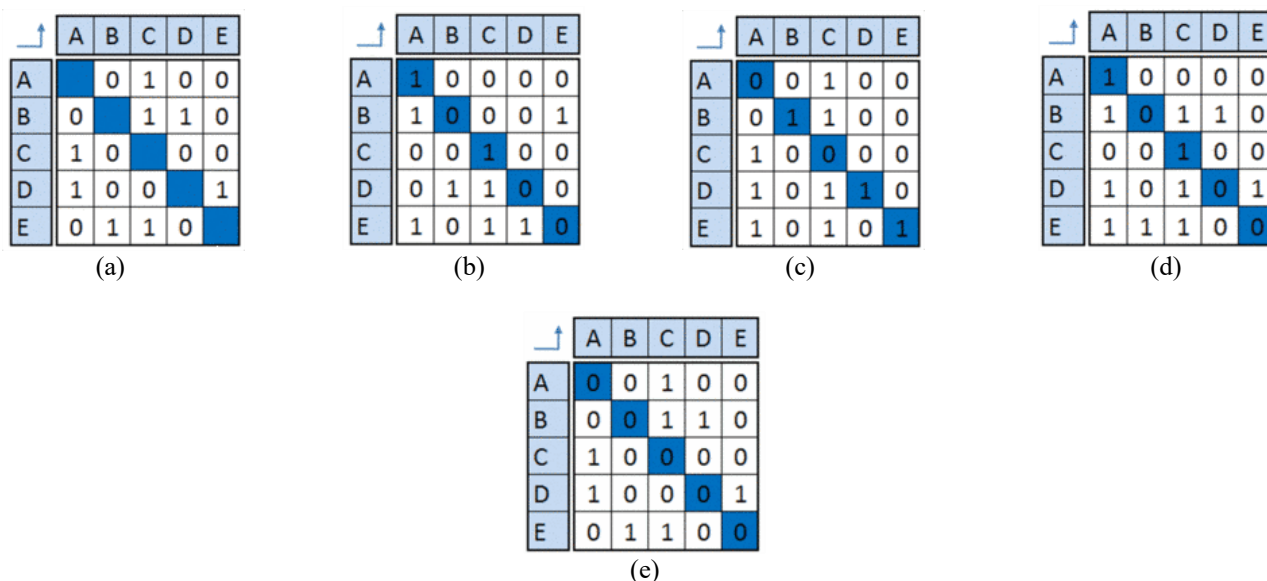


Fig 7.2.2.1: (a) DSM di partenza, (b) DSM^2 , (c) DSM^3 , (d) DSM^4 , (e) DSM^5

L'esempio seguente mostra, tuttavia, che questo metodo presenta alcuni limiti

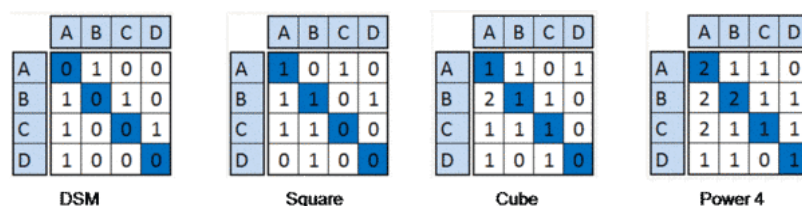


Fig. 7.2.2.2: esempio che illustra i limiti del procedimento

La matrice quadrata risultante dal DSM originale (come si vede in figura 7.2.2.2 lato sinistro) possiede due voci sulla diagonale, che indicano che gli elementi A e B formano un ciclo di feedback. Cubando la matrice si ottengono tre voci sulla diagonale che mostrano che gli elementi A, B e C formano un altro ciclo di feedback. La carenza di questo approccio può essere vista se viene calcolata la quarta potenza della matrice (matrice sul lato destro). A causa di un ciclo di feedback che comprende tutti e quattro gli elementi della struttura iniziale (A, B, C e D), tutte le celle diagonali mostrano voci non vuote; tuttavia, le celle diagonali degli elementi A e B contengono ciascuna il valore "2". Questo risultato si ottiene perché questi elementi formano un proprio ciclo di feedback (comprendente solo questi due elementi), che viene percorso due volte alla quarta potenza della matrice. Ciò dimostra che i cicli di feedback completi di strutture più grandi non possono essere

analizzati con questo approccio, perché i cicli di feedback contenenti gli stessi elementi non possono essere ricondotti distintamente ai loro elementi impliciti. Il metodo consente solo la determinazione dell'esistenza, ma non l'identificazione dettagliata dei circuiti di feedback. Inoltre, questo approccio non sempre consente di acquisire la lunghezza e la quantità dei circuiti di feedback esistenti. Se in una matrice si devono identificare cicli di retroazione che si estendono su sei elementi, i cicli di retroazione comprendenti due elementi vengono elaborati tre volte e i cicli di retroazione comprendenti tre elementi vengono elaborati due volte quando la matrice viene elevata alla sesta potenza. Tutti questi cicli di feedback vengono aggiunti ai valori risultanti nelle celle diagonali del DSM, rendendo quasi impossibile l'identificazione di uno specifico ciclo di feedback.

7.2.3 Il metodo della matrice di raggiungibilità:

La matrice di raggiungibilità è un DSM binario con gli elementi diagonali uguali a “UNO”. Il metodo richiede di trovare una scomposizione gerarchica a più livelli per la matrice. Il livello più alto di questa gerarchia è composto da tutti gli elementi che non richiedono input o sono indipendenti da tutti gli altri elementi della matrice. Due elementi qualsiasi allo stesso livello della gerarchia non sono collegati tra loro o fanno parte dello stesso circuito a quel livello. Una volta identificato l'insieme di elementi di livello superiore, gli elementi nell'insieme di livello superiore e le loro corrispondenti connessioni da/a vengono rimossi dalla matrice lasciandoci con una sottomatrice che ha il proprio insieme di livello superiore. L'insieme di livello superiore di questa sottomatrice sarà l'insieme di secondo livello della matrice originale. Procedendo in questo modo si possono individuare tutti i livelli della matrice.

7.3 “Tearing” ovvero “Lacerazione” di un DSM:

Una volta identificato un sottoinsieme di elementi accoppiati in un DSM (vedi Sequenza del DSM), lo strappo o lacerazione è un modo per tentare di determinare una sequenza per gli elementi in questo sottoinsieme. Lo strappo è il processo di scelta dell'insieme di segni di feedback che, se rimossi dalla matrice (e quindi la matrice viene nuovamente partizionata), renderanno la matrice triangolare superiore. I segni che rimuoviamo dalla matrice si chiamano “lacrime”. Identificare quegli “strappi” che danno luogo a una matrice triangolare superiore significa che abbiamo identificato l'insieme di ipotesi che devono essere fatte per avviare le iterazioni del processo di progettazione quando si incontrano attività accoppiate nel processo. Fatte queste ipotesi non è necessario effettuare ulteriori stime. Non esiste un metodo ottimale per lo strappo, ma consigliamo l'uso di due criteri quando si prendono decisioni in merito allo strappo:

- Numero minimo di strappi: la motivazione alla base di questo criterio è che gli strappi rappresentano un'approssimazione o una prima ipotesi da utilizzare; preferiremmo ridurre il numero di queste ipotesi utilizzate.
- Limitare gli strappi ai blocchi più piccoli lungo la diagonale: la motivazione dietro questo criterio è che se devono esserci iterazioni all'interno di iterazioni (cioè, blocchi all'interno di blocchi), queste iterazioni interne vengono eseguite più spesso. Pertanto, è auspicabile limitare le iterazioni interne a un numero limitato di compiti.

7.4 Simulazione del modello di iterazione:

In questa ultima fase si seleziona il blocco DSM caratterizzato dalle attività che costituiscono un ciclo o loop. Si effettua una decisione sulla sequenza di esecuzione all'interno del blocco in esame per iniziare l'iterazione: questa decisione può essere arbitraria, ciò che è importante è che il processo di “strappo” sia eseguita correttamente. A questo punto si sostituiscono i segni X nel blocco di iterazione con i valori di probabilità di rilavorazione: le probabilità di rielaborazione sono il frutto di una disamina opportunamente studiata da parte di esperti del settore. Per progetti già eseguiti in passato i valori sono di facile costruzione; per i progetti nuovi la cosa è più complessa perché richiede da parte degli esperti uno sforzo di previsione e di analisi delle variabili in gioco. Il passaggio successivo è eseguire la modellazione dell'iterazione ed il metodo migliore è la simulazione anche e non è l'unico. La simulazione permette di ottenere la durata dell'iterazione del blocco ed è quindi possibile aggiornare il DSM con la nuova durata.

7.4.1 Descrizione del modello proposto dal MIT di Boston e in particolare del lavoro di Browning ed Eppinger del 2001:

Il modello matematico che ora affrontiamo è stato sviluppato e implementato nei fogli Excel con macro messi a disposizione degli studenti affinché familiarizzino con lo strumento e comprendano come utilizzarlo. Tale modello caratterizza il processo produttivo come una rete di attività che si scambiano prodotti finiti. Se un'attività svolge un lavoro e produce un output basato su input o ipotesi, una modifica di uno dei due può comportare una rielaborazione per l'attività. La probabilità di rilavorazione è indicata nella matrice DSM_{xy1} . La rilavorazione può avere un impatto variabile su un'attività: alcuni cambiamenti possono essere assorbiti da un'attività robusta con un impatto minimo mentre la conseguenza della modifica di altri input può essere più grave. Gli impatti per le rilavorazioni sono indicati nella matrice DSM_{xy2} . Il modello stabilisce una regola di lavoro tra le attività i cui punti essenziali sono i seguenti:

- 1) un'attività non può iniziare finché non ha ricevuto tutti i suoi input dalle attività precedenti o “a monte” (quelle elencate prima nel vettore di sequenziamento, V);
- 2) un'attività può fare ipotesi sui suoi input da attività successive o “a valle”. Pertanto, le attività possono lavorare in contemporanea solo se quella a valle non dipende da quella a monte per gli input;
- 3) solo le attività adiacenti in V possono lavorare contemporaneamente.

La sequenza di attività del DSM e la politica di lavoro che regola le loro interazioni e sovrapposizioni determinano un'architettura di processo. Questa politica di lavoro si svolge come segue: l'attività più a monte che richiede lavoro viene identificata e resa attiva. Poi, le successive attività a valle vengono controllate per verificare la dipendenza dall'attività attiva. Se l'attività successiva: 1) ha del lavoro da svolgere e 2) non dipende dall'attività attiva, anche questa attività susseguente viene attivata. Ogni attività successiva che richiede lavoro viene controllata in modo analogo per verificare la dipendenza dalle attività attive fino a quando non viene trovata un'attività dipendente, a quel punto è stato determinato l'insieme attuale di attività attive. Quindi solo le attività consecutive (di quelle che richiedono lavoro) possono essere attive, rendendo l'insieme dipendente dall'architettura di processo. Le attività finite e le ipotesi sulle attività successive sono entrambe ignorate nel determinare l'insieme attivo. Nel caso di attività accoppiate, la politica di lavoro di base implica che una delle attività vada per prima e l'altra aspetti. Inoltre, implica che le attività a valle si fermeranno e aspetteranno quando le attività a monte da cui dipendono saranno riattivate a causa di una rielaborazione.

7.4.2 Sommario del modello e delle variabili di simulazione:

C: processo cumulativo di costo per una data esecuzione

S: processo cumulative di durata per u a data esecuzione

t: tempo fino al successivo evento

n: numero delle attività di processo

W_n “vettore lavoro”: un vettore di lunghezza n con un valore di entrata [0,1] per ciascuna attività, che indica il lavoro che deve essere fatto su quella attività; inizialmente l'insieme degli 1 indica che rimane da fare il 100% del lavoro per ciascuna attività

WN_n “vettore lavoro attuale”: un vettore di lunghezza n con un valore di ingresso booleano per ciascuna attività, che indica le attività attive per il corrente stato del sistema

IC_n “vettore curva di incremento”: un vettore di lunghezza n con un valore di ingresso [0,1] per ciascuna attività che indica la percentuale richiesta della durata e del costo della attività originale per le seconde e successive esecuzioni dell'attività

DSM_{nkk} “DSM (matrice nxn con la terza dimensione k)”:

- Dimensione k=1: probabilità di rilavorazione [0,1]
Per ingressi sopra la diagonale principale dove la riga $i <$ della colonna j , la probabilità dell'attività j di causare una rilavorazione per l'attività i ; per ingressi al di sotto della diagonale principale dove $i > j$ la probabilità dell'attività j di causare un secondo ordine di rilavorazione per l'attività i dopo la rilavorazione dell'attività j
- Dimensione k=2: impatto di rilavorazione [0,1]
Il lavoro aggiuntivo (in termini di incremento di W) che la rilavorazione dovrebbe procurare

7.4.3 Algoritmo qualitativo del codice di simulazione:

1. Inizializzare le variabili del modello dagli input allo stato 0.
2. Campionare casualmente la durata e il costo di ciascuna attività (con appropriata correlazione, utilizzando la tecnica di campionamento dell'ipercubo latino).
3. Per lo stato attuale del sistema:
 - A) Determinare l'insieme delle attività attive, in base alla politica del lavoro:
 - I. imposta tutto $WN = FALSE$.
 - II. Trovare la maggior parte dell'attività a monte, i , con lavoro incompiuto – ovvero dove $W(i) > 0$. Impostare $WN(i) = VERO$.
 - III. Passa attraverso le attività successive.
 - i. Se l'attività successiva ha un lavoro non completato E non dipende da un'attività a monte non completata, impostare la relativa voce WN su $TRUE$.

ii. In caso contrario è stato trovato l'insieme delle attività attive completo (interrompere il controllo delle attività).

B) Calcolare il tempo fino all'evento successivo (termina l'attività attiva più breve), t

C) Lavoro su attività attive (decremento W); incrementare il tempo cumulativo di t ; incrementare il costo cumulativo

D) verificare eventuali rilavorazioni generate dall'attività completata j

I. Esamina la colonna j nel DSM1 (sopra l'attività j) per una potenziale iterazione.

Valuta ogni possibilità rispetto a un numero casuale; se l'iterazione avviene per l'attività i , allora:

i. Aggiungi la rielaborazione appropriata (data in DSMij2, modificata da ICi) a W_i . Ma se $W_i > 1$ adesso (perché l'attività i non era terminata), riduci W_i a 0,9 per evitare che il lavoro si espanda oltre l'ambito originale e per rappresentare qualche miglioramento.

ii. Esamina la colonna i nel DSM1 (sotto l'attività i) per potenziali rielaborazioni di secondo ordine.

iii. Valutare ciascuna possibilità (dove $W < 1$) rispetto a un numero casuale; se si verifica una rilavorazione per l'attività k , allora:

- aggiungere la rielaborazione appropriata (data nel DSMki2, modificata da ICk) a W_k . Ma se $W_k > 1$ ora (perché l'attività k non è stata completata), ridurre W_k a 0,9 per evitare che il lavoro si espanda oltre lo scopo originale e per rappresentare qualche miglioramento.

4. Se una qualche attività ha più lavoro da svolgere, incrementare lo stato del sistema e ripetere 3.

5. Uscita C e S

7.4.4 Spiegazione dell'algoritmo di simulazione:

il modello utilizza una simulazione a eventi discreti per calcolare le distribuzioni di durata e costo per un dato insieme di input. Ogni esecuzione della simulazione inizia dallo stato 0 del sistema (valori iniziali), con una durata e un costo (con correlazione) campionati casualmente per ogni attività. Inizialmente ogni attività ha il 100% del lavoro da svolgere, come indicato nel vettore "lavoro da svolgere" W . Quando $W = 1$ l'intera attività i rimane da svolgere; quando $W = 0$ l'attività i è completa. In ogni stato del sistema il modello determina l'insieme delle attività attive in base alla regola di lavoro tra le attività descritte in precedenza. L'attività più breve dell'insieme attivo sarà la prossima a terminare e a generare un output quindi la sua durata determina il tempo fino all'evento successivo. Una volta terminata un'attività il tempo dell'evento viene aggiunto alla durata cumulativa e il costo del lavoro svolto su tutte le attività attive viene aggiunto al costo cumulativo. Frazioni adeguate di W vengono sottratte alle attività attive e il modello controlla eventuali rilavorazioni causate dall'output dell'attività completata. Per determinare la rilavorazione si effettua un controllo probabilistico per le potenziali iterazioni e per le rilavorazioni di secondo ordine utilizzando le probabilità inserite in DSMxy1 e DSMxy2. Quando tutte le attività sono completate tutte le voci di W sono uguali a 0 e la simulazione produce il costo e la durata cumulativi come C e S rispettivamente.



PARTE 2

In questa seconda parte si illustrano gli strumenti informatici a disposizione per l'analisi dei dati della matrice DSM e che il lettore, che voglia approfondire la materia, deve conoscere e saper utilizzare. Viene presentato lo strumento messo a disposizione dai ricercatori del MIT di Boston ossia un file Excel con Macro con all'interno 11 schede di lavoro. Queste schede di lavoro, che rappresentano una sorta di percorso da seguire per l'ottenimento del processo ottimizzato, vengono analizzate anche alla luce dell'interpretazione del codice VBA che implementa i fogli di lavoro stessi con le sue macro. Lo strumento del MIT è stato aggiornato con l'aggiunta di una scheda di lavoro denominata "Activities-Parameters". La programmazione effettuata del codice VBA, delle macro in esso contenute, è un ulteriore tassello che rende lo strumento estremamente interessante anche per futuri sviluppi. Dai testi letti, dagli articoli e dalle informazioni reperite l'approccio proposto con l'inserimento del foglio di lavoro "Activities-Parameters" rende la procedura di costruzione della matrice DSM maggiormente adatta alla condivisione di informazioni tra membri di un team concentrato sulla formazione della matrice stessa. Il Consensus Panel di esperti allestito per raggiungere un determinato obiettivo può trovare questa soluzione di grande utilità in quanto evita l'approccio diretto utile solo a coloro che conoscono profondamente lo strumento DSM. Viene presentato inizialmente un software curato dai ricercatori dell'Università di Cambridge davvero interessante: la sua analisi è di estrema utilità per arricchire le possibilità di coloro che utilizzeranno queste pagine, magari come punto di partenza per ulteriori analisi. Il software è modulabile ed espandibile nelle sue funzionalità. Nella parte finale si presenta il plugin "Adjacency Matrix Exporter" scaricabile dalla piattaforma Obsidian.md in grado di generare due tipi di matrici di adiacenza numeriche programmato in linguaggio Typescript: in bibliografia sono suggeriti testi di approfondimento su questo specifico linguaggio di programmazione.

8 Software CAM Ver.2 (a cura dei ricercatori dell'Università di Cambridge)

Il Cambridge Advanced Modeller (CAM) è uno strumento software per modellare e analizzare le dipendenze e i flussi di cambiamento in sistemi complessi, come prodotti, processi e organizzazioni. Per i professionisti, si è trattato di una risposta alle esigenze dell'industria fornendo un modo sistematico di acquisire e analizzare molteplici livelli di complessità. Il CAM fornisce un mezzo per raggiungere questo obiettivo introducendo diversi strumenti. Per il mondo accademico, il CAM può essere una preziosa risorsa di strumenti di modellazione e analisi, un diagrammatore, uno strumento di simulazione e uno strumento DSM, che incorpora la conoscenza proveniente da precedenti applicazioni industriali con i nuovi approcci di modellazione. È gratuito per la ricerca, l'insegnamento e la valutazione. Le caselle degli strumenti in CAM forniscono diverse notazioni di modellazione e metodi di analisi. Il CAM può essere configurato per sviluppare nuove notazioni di modellazione specificando i tipi di elemento e connessione consentiti. Un'architettura modulare consente di aggiungere nuove funzionalità come codici di simulazione.

8.1 Come costruire un modello DSM

Esistono due modi per creare DSM in CAM. È possibile creare un modello da zero come DSM, utilizzando la casella degli strumenti DSM, oppure si può creare una vista DSM per visualizzare dati già esistenti, creati utilizzando una vista diagramma. Nella schermata delle raccolte (che viene visualizzata dopo aver avviato CAM) fare clic sul pulsante "DSM Collection" nella fascia "Insert" nella scheda "Home" come mostrato in figura 8.1.1. Verrà creata una nuova raccolta DSM che comprende un modello DSM.

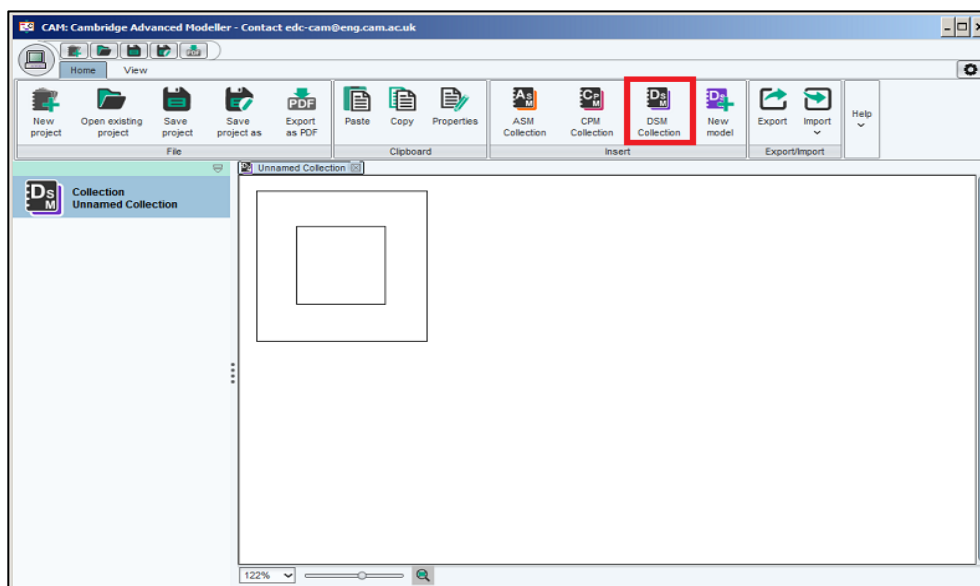


Fig. 8.1.1: schermata di visualizzazione dopo aver premuto il pulsante “DSM Collection”

Fare doppio clic sulla miniatura del modello DSM visualizzata nel riquadro di destra per visualizzare la scheda del modello DSM. Poiché la matrice è vuota dopo la creazione, appare come un foglio di lavoro "vuoto". Bisogna seguire le istruzioni di seguito riportate nella sezione "Creazione di nuovi elementi (righe/colonne)" per aggiungere elementi e dipendenze alla nuova matrice. Innanzitutto, aprire il modello che si desidera visualizzare come DSM. Quindi creare una nuova vista di tipo DSM e seguire le istruzioni riportate di seguito per utilizzare la funzionalità DSM nella nuova vista.

8.2 Importazione di DSM da CSV (Excel)

Innanzitutto, il file CSV deve avere lo stesso formato di quello creato dall'esportazione DSM. Seguire i passaggi seguenti per importare DSM.

1. Creare una nuova raccolta DSM o aprire una raccolta DSM salvata.
2. Aprire il foglio del modello DSM.
3. Fare clic sul pulsante "Importa" nella banda "Export/Import" nella scheda "Home" che richiama un elenco a comparsa con diverse opzioni di importazione.
4. Seleziona l'opzione "Importa DSM da CSV" e seleziona il file CSV che desideri importare. Verrà visualizzata la finestra di dialogo di importazione che mostra un'anteprima del layout dei dati da importare. Assicurarsi che i dati siano visualizzati correttamente. (Nella finestra di dialogo di importazione sono disponibili opzioni per regolare la riga e la colonna iniziale, il che aiuta a rimuovere righe e colonne vuote prima dell'inizio dei dati, se presenti.)
5. Selezionare "Importa" e i dati verranno importati esattamente nel layout mostrato nel file CSV.

Tenere presente che l'importazione DSM è molto sensibile al formato del file; i dati da importare devono seguire esattamente il formato creato dall'esportazione DSM.

8.3 Creazione di nuovi elementi (righe/colonne):

selezionare il pulsante "DSM Element", come mostrato in figura 8.3.1, nella fascia "Palette" della scheda "DSM Modelling".

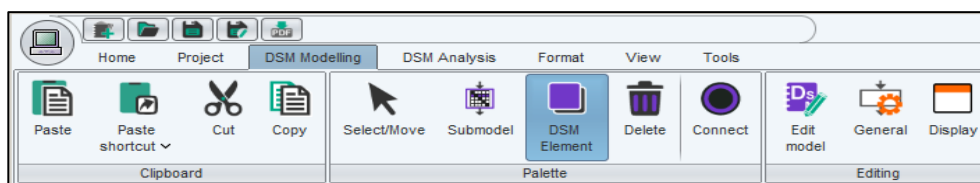


Fig. 8.3.1: pulsante "DSM Element"

Fare clic una volta nella riga di intestazione (o nella colonna di intestazione) nella posizione in cui si desidera creare il nuovo elemento. Per una matrice vuota (ad esempio un DSM appena creato) fare clic in un punto qualsiasi del foglio di lavoro vuoto per creare il primo elemento. Tenere presente che ogni successivo clic del mouse creerà un elemento aggiuntivo, finché non si seleziona un altro strumento (ad esempio lo strumento di spostamento, che viene visualizzato come cursore del mouse, il cui pulsante si trova all'estrema sinistra della banda "Palette").

Creazione di dipendenze: utilizzare lo strumento di connessione (situato all'estrema destra della banda "Palette") e fare clic nel riquadro della matrice principale per creare dipendenze

Eliminazione di elementi e dipendenze: Utilizza lo strumento Elimina e fai clic una volta per eliminare un elemento o una dipendenza. Torna allo strumento di spostamento una volta terminata l'eliminazione.

Esportazione DSM: la matrice DSM può essere esportata in diversi formati, come mostrato in figura 8.3.2. Può essere esportata graficamente come PDF e PNG. I dati DSM possono essere esportati come CSV (Excel). Fare clic sul pulsante "Esporta" nella banda "Export/Import" nella scheda Home e verrà visualizzata la finestra di dialogo delle opzioni di esportazione. Seleziona il formato richiesto e puoi scegliere di aprire immediatamente l'esportazione o salvarla come file CSV.

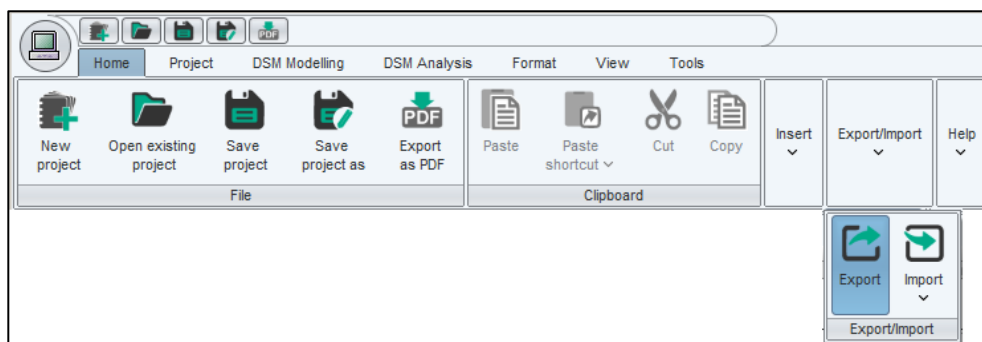


Fig. 8.3.2: esportazione DSM

L'esportazione come PDF può essere eseguita anche utilizzando il pulsante "Export as PDF" nella fascia "File" della scheda "Home" o utilizzando il pulsante "PDF" nella barra degli strumenti in alto.

8.4 Analisi:

Sono inclusi algoritmi DSM standard per riorganizzare automaticamente un DSM, o un DSM parziale, per evidenziare o scoprire caratteristiche strutturali. Questi sono:

- Partition: utilizza un algoritmo di ricerca in loop per trovare un ordinamento tale che il maggior numero possibile di segni sia sotto la diagonale principale e la "dimensione" dei loop di feedback sia ridotta al minimo. Ciò può corrispondere ad un "ordinamento ottimale" di un processo.
- Cluster: raggruppa automaticamente la matrice (o cluster) in sottocluster fortemente connessi.
- Band: raggruppa gli elementi successivi in un cluster se è possibile tentarli contemporaneamente.
- Flatten cluster: separa o appiattisce il cluster già creato.

Gli algoritmi sono disponibili selezionando lo strumento appropriato nella fascia "DSM Algorithms" nella scheda "DSM Analysis" così come è visibile in figura 8.4.1, quindi facendo clic sulla matrice o sul cluster specifico in cui verrà applicato l'algoritmo.

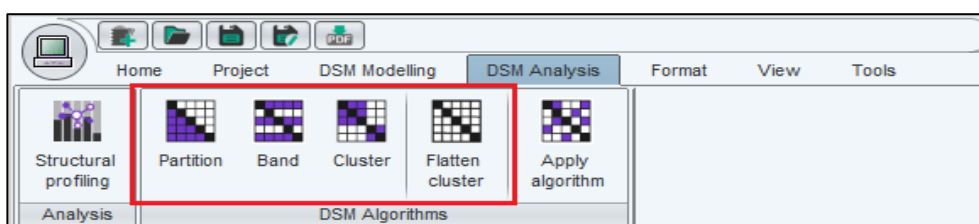


Fig. 8.4.1: fascia DSM Algorithms

È inoltre possibile applicare gli algoritmi sulla matrice in una finestra pop-up separata utilizzando il pulsante "Applica algoritmo" ("Apply algorithm") nella fascia "Algoritmi DSM" nella scheda "Analisi DSM". Ciò consente all'utente di provare diversi algoritmi sulla matrice selezionata prima di aggiornare o salvare la matrice con le modifiche.

8.5 Risultati:

Diversi algoritmi DSM possono essere applicati alle matrici per ottenere un ordine ottimale di elementi o per consentire il raggruppamento per identificare elementi fortemente connessi o per scoprire la concorrenza, a seconda del tipo di matrice e dell'analisi richiesta. I risultati dell'analisi DSM possono essere salvati nel modello stesso o come una vista diversa all'interno del modello: si da una visione di questo fatto nella figura 8.5.1. Durante l'applicazione degli algoritmi all'utente verrà visualizzata una finestra di dialogo con le seguenti opzioni:

- salva le modifiche come una nuova vista: i risultati dell'algoritmo verranno salvati come una vista diversa che sarà elencata nella cartella "Visualizzazioni" all'interno del modello aperto nell'albero di esplorazione del progetto nel riquadro laterale sinistro;
- accetta le modifiche nella visualizzazione corrente: i risultati dell'algoritmo verranno salvati nel modello originale.

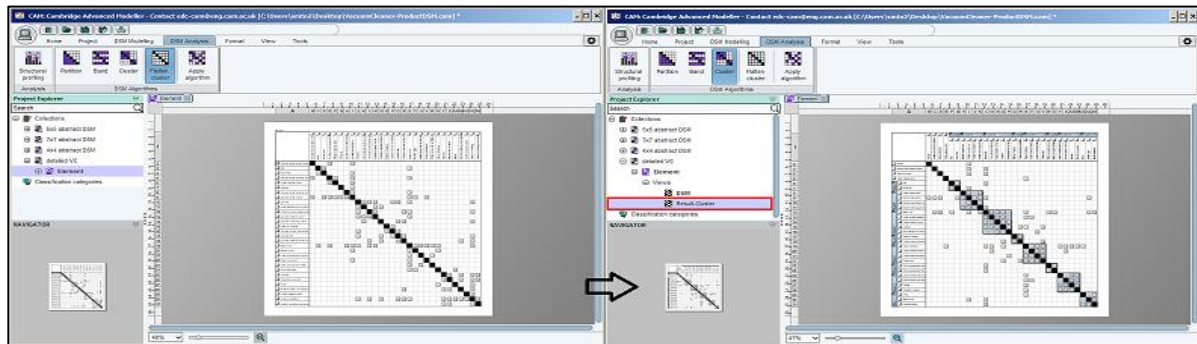


Fig 8.5.1: schermata a confronto dopo le due opzioni di salvataggio

9 Fogli Excel con macro: DSM_Program-V2.1.xls (a cura del team di ricerca del MIT di Boston)

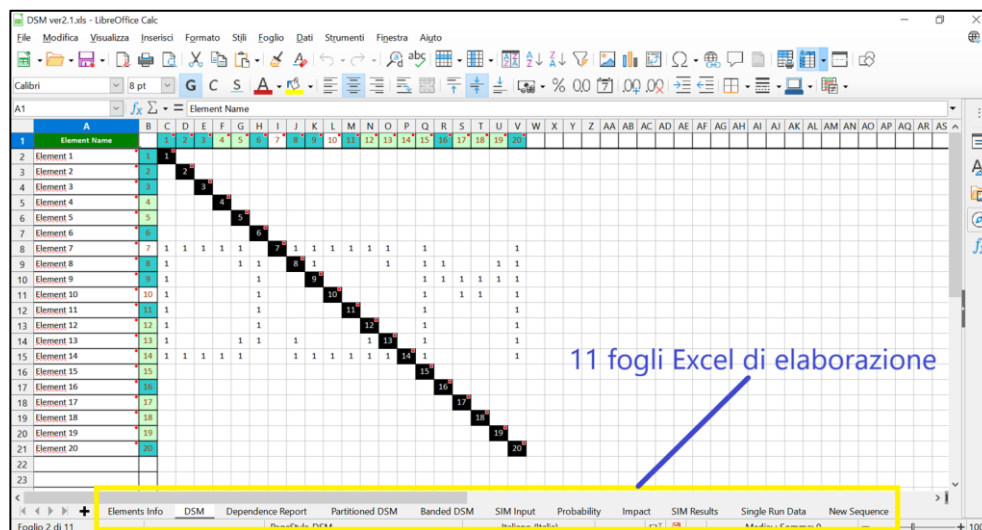


Fig. 9.1: schermata iniziale del foglio Excel con indicazione degli 11 fogli di elaborazione

9.1 Introduzione a VBA:

Visual Basic for Applications (VBA) è un linguaggio di programmazione creato da Microsoft e integrato nelle applicazioni della suite Office, come Excel, Word e Access. L'introduzione di VBA negli anni '90 ha rappresentato una svolta importante nell'automazione e nell'estensione delle funzionalità all'interno di queste applicazioni. Inizialmente, VBA è stato implementato con Excel 5.0 nel 1993 e successivamente esteso a Word, Access e PowerPoint. La sua origine come estensione di Visual Basic 3.0 ha contribuito a rendere VBA un linguaggio di scripting potente, adatto anche a utenti senza una formazione approfondita in programmazione. VBA ha reso accessibili le funzionalità di scripting a un vasto pubblico, consentendo agli utenti di automatizzare compiti ripetitivi e di creare macro per personalizzare le applicazioni Office in base alle proprie esigenze. Con il suo approccio orientato agli oggetti, VBA consente agli sviluppatori e agli utenti di interagire direttamente con gli oggetti interni delle applicazioni Office, come celle, paragrafi e record. Nel corso degli anni, VBA ha continuato a evolversi, integrandosi sempre di più nelle nuove versioni di Office con nuove funzionalità e miglioramenti. Tuttavia, con l'introduzione di Office 2021, Microsoft ha annunciato la fine del supporto per VBA nelle versioni future, anche se rimane supportato nelle versioni più recenti di Office. Nonostante l'emergere di alternative come l'Office JavaScript API, VBA conserva la sua rilevanza e popolarità nella comunità degli sviluppatori grazie alla sua facilità d'uso e alla vasta base di utenti. La comunità VBA continua a crescere, fornendo risorse online, forum di discussione e supporto per coloro che desiderano sfruttare appieno le potenzialità di questo linguaggio all'interno delle applicazioni Office. In sintesi, VBA rappresenta un importante strumento per l'automazione e la personalizzazione nell'ambito delle applicazioni Microsoft Office.

9.2 Foglio “Element Info”:

Questo foglio di lavoro contiene informazioni dettagliate sugli elementi inclusi nella matrice DSM. Ecco un riassunto delle operazioni che coinvolgono la scheda in oggetto:

1. **Elementi e informazioni associati:** Ogni riga rappresenta una attività di processo, e alcune colonne contengono informazioni specifiche su ciascuna attività.
2. **Colonne utilizzate nel codice:**
 - Colonna 1: viene indicato l'ID o l'indice dell'elemento.
 - Colonna 2: contiene commenti o informazioni aggiuntive sull'elemento.

- Colonna 3: contiene il nome dell'elemento.
 - Colonna 5: rappresentare il livello di completamento dell'attività, espressa in percentuale.
3. **Caricamento delle informazioni:** La procedura `Update_DSM()` utilizza le informazioni contenute nella scheda per aggiornare la matrice DSM. Vengono considerati elementi precedentemente esistenti, e le informazioni vengono utilizzate per formattare il foglio di lavoro.
 4. **Restaurare le relazioni precedenti:** Se un elemento nella scheda "Elements Info" ha relazioni precedentemente definite (colonna 1 non vuota), queste relazioni vengono ripristinate nella matrice DSM durante l'aggiornamento.

9.3 Foglio "DSM":

La scheda "DSM" è il luogo principale dove viene rappresentata la matrice DSM (Design Structure Matrix) all'interno dell'applicazione.

Ecco un riassunto delle operazioni che coinvolgono la scheda "DSM" nel codice fornito:

1. **Rappresentazione della matrice DSM:** La matrice DSM sembra essere rappresentata nella parte superiore sinistra del foglio di lavoro "DSM". Ogni cella della matrice rappresenta l'interazione tra due elementi del sistema. La presenza di un valore nella cella potrebbe indicare una dipendenza o un'interazione tra gli elementi corrispondenti.
2. **Aggiornamento della matrice DSM:** La procedura `Update_DSM()` aggiorna la matrice DSM sulla base delle informazioni presenti nella scheda "Elements Info". Vengono utilizzate le informazioni sugli elementi, i loro nomi e lo stato di completamento per formattare la matrice e ripristinare le relazioni precedentemente definite.
3. **Operazioni di modifica sulla matrice DSM:**
 - **Cancellazione delle relazioni:** La procedura `Clear_DSM()` cancella tutte le relazioni presenti nella matrice DSM, portando la matrice a uno stato iniziale senza interazioni definite.
 - **Scambio di elementi:** C'è un'opzione nel menu ("Swap two elements") che consente di scambiare due elementi nella matrice DSM.
 - **Tearing e Untearing:** Ci sono opzioni nel menu ("Tear DSM Mark", "Untear Previous Mark", "Tear DSM Column") che coinvolgono la manipolazione delle relazioni nella matrice DSM.
4. **Generazione di report e simulazione:** Ci sono opzioni nel menu ("Generate dependence report", "Simulate DSM") che coinvolgono la generazione di report sulle dipendenze e la simulazione della matrice DSM.

In sintesi, la scheda "DSM" sembra essere il luogo principale dove viene visualizzata e manipolata la matrice DSM, e molte delle operazioni coinvolgono la gestione delle relazioni e la rappresentazione visiva delle interazioni tra gli elementi del sistema.

9.4 Foglio "Dependence Report":

La subroutine "Dependence_Report" è progettata per analizzare una matrice di dipendenza rappresentata nel foglio di lavoro "DSM" (Design Structure Matrix) e generare un rapporto che indica quali elementi dipendono da altri.

Dall'analisi del codice VBA si ricava che in questo foglio si possono fare le seguenti operazioni:

1. **Pulizia del Report Precedente:**
 - Prima di generare un nuovo rapporto, la subroutine cancella eventuali dati precedenti presenti nella colonna B del foglio di lavoro "Dependence Report".
 - La cella B1 viene quindi etichettata con il titolo "Depends on".
2. **Generazione del Nuovo Rapporto:**

- La colonna A del foglio di lavoro "Dependence Report" viene riempita con i nomi degli elementi, ottenuti dal foglio "Elements Info".
- Utilizzando una doppia iterazione sui dati nella matrice nel foglio "DSM", la subroutine cerca le dipendenze tra gli elementi.
- Se l'elemento i dipende dall'elemento j (rappresentato dal valore "1" nella cella corrispondente nel foglio "DSM"), il nome dell'elemento j viene aggiunto alla cella corrispondente nella colonna B del foglio "Dependence Report". Viene anche aggiunto un separatore ", " per distinguere chiaramente i nomi degli elementi.

In definitiva, il risultato è un rapporto chiaro delle dipendenze tra gli elementi, che può essere utile per comprendere la struttura del progetto o del sistema rappresentato dalla matrice DSM. Il rapporto indica, per ciascun elemento, da quali altri elementi dipende. Questo tipo di analisi può essere cruciale per la progettazione e la gestione dei sistemi complessi.

9.5 Foglio “Partitioned DSM”

La scheda "Partitioned DSM" rappresenta una versione della matrice DSM (Design Structure Matrix) che è stata suddivisa o partizionata in gruppi di elementi. La partizione è il risultato di un processo di analisi e raggruppamento degli elementi in base alle loro dipendenze reciproche.

La subroutine associata alla scheda "Partitioned DSM" nel codice VBA è chiamata "Partition", e il suo scopo principale è quello di eseguire la partizione della matrice DSM. Vediamo una descrizione di alto livello di cosa fa la subroutine:

1. Preparazione dell'Ambiente:

- Viene effettuata la pulizia e la formattazione del foglio di lavoro "New Sequence", che viene utilizzato per visualizzare la sequenza degli elementi dopo la partizione.
- Anche la scheda "Partitioned DSM" viene preparata con una formattazione di base.

2. Inizializzazione delle Variabili:

- Vengono inizializzate alcune variabili, tra cui vettori (arrays) come `taken`, `temp`, e `C`, che vengono utilizzati durante il processo di partizione.

3. Ciclo di Partizione:

- Viene eseguito un ciclo che continua finché tutti gli elementi non sono stati assegnati a un livello di partizione.
- Durante ogni iterazione del ciclo, viene analizzato ogni elemento per determinare se è possibile assegnarlo a un livello di partizione.
- Se un elemento può essere assegnato, viene marcato come "preso" (`taken`) e spostato nel foglio "New Sequence" al livello appropriato.

4. Visualizzazione dei Risultati:

- Dopo la partizione, i risultati vengono visualizzati sul foglio "Partitioned DSM". Viene evidenziato il raggruppamento degli elementi in base ai livelli di partizione.

5. Aggiornamento delle Dipendenze:

- Vengono aggiornate le dipendenze tra gli elementi sulla base della partizione, in modo che la matrice rifletta correttamente la struttura della partizione.

La scheda "Partitioned DSM" e la subroutine "Partition" sono utili strumenti per analizzare la complessità di un sistema o di un progetto, suddividendo gli elementi in gruppi più gestibili in base alle loro relazioni e dipendenze. Questo può facilitare la comprensione e la gestione di progetti complessi.

9.6 Foglio “Banded DSM”:

Nel contesto della subroutine `Banding_Partitioned_DSM`, le attività evidenziate rappresentano il "set più a monte" di attività che possono essere eseguite simultaneamente in ciascun passo temporale. Questo set di attività è stato identificato in base alle dipendenze tra le attività presenti nella Design Structure Matrix (DSM) partizionata. Quando un'attività viene evidenziata con un colore di sfondo specifico nel foglio "Banded DSM", significa che fa parte del set di attività che possono essere

svolte contemporaneamente senza dipendenze tra loro durante un determinato passo temporale. Questa operazione di "banding" è utile per visualizzare e comprendere le relazioni di dipendenza tra diverse attività in un progetto o processo. In sintesi, le attività evidenziate rappresentano un gruppo di attività che possono essere eseguite simultaneamente in base alla loro organizzazione nella DSM partizionata, contribuendo a una migliore comprensione delle relazioni di dipendenza nel contesto del progetto.

9.7 Foglio "SIM Input":

gestisce l'input di dati per le durate delle attività e fornisce funzionalità per cancellare e accettare gli input. Questa di seguito è una breve spiegazione delle procedure presenti nel codice:

1. **SIM_Input:**

Questa subroutine popola la scheda "SIM Input" con i dati delle attività. Le informazioni includono il numero di sequenza (`New_Seq(i)`), i parametri dell'attività (`Parameter(New_Seq(i))`), il numero dell'attività (`Worksheets("DSM").Cells(i + 1, 2)`), e il nome dell'attività (`Worksheets("DSM").Cells(i + 1, 1)`).

2. **Clear_Sim_Input:**

Questa subroutine cancella il contenuto delle colonne da "C" a "F" sulla scheda "SIM Input". In altre parole, cancella i dati delle durate delle attività precedentemente inseriti.

3. **Accept_Sim_Input:**

Questa subroutine sembra navigare verso la scheda "Probability" quando viene eseguita. Tuttavia, questa porzione specifica del codice non contiene il codice effettivo per il cambio di scheda. Potrebbe essere necessario aggiungere o cercare il codice mancante nel contesto più ampio dell'applicazione. Queste procedure sembrano essere legate all'input di dati per simulazioni e analisi di probabilità. L'utente può inserire durate delle attività nella scheda "SIM Input", cancellare gli input con "Clear_Sim_Input" e poi procedere ad altre fasi, come visualizzare la probabilità. La scheda "SIM Input" viene utilizzata per inserire i dati necessari per la simulazione. La procedura `SIM_Input` in particolare si occupa di popolare questa scheda con le informazioni relative alle attività. Di seguito elenco una serie di operazioni che vengono fatte:

1. **Iterazione sulle Attività:**

- Un ciclo `For` viene utilizzato per iterare attraverso tutte le attività (`elementNum` indica il numero di attività).
- Per ogni attività, le informazioni vengono estratte dalla scheda "DSM" utilizzando le celle nelle colonne 1 e 2:
 - `Worksheets("DSM").Cells(i + 1, 2)`: Numero dell'attività
 - `Worksheets("DSM").Cells(i + 1, 1)`: Nome dell'attività
- Queste informazioni vengono quindi inserite nella scheda "SIM Input" nelle colonne 1 e 2 rispettivamente.

2. **Cancellazione del Contenuto Esistente:**

- Prima di inserire i nuovi dati, la procedura cancella il contenuto delle colonne da "C" a "F" sulla scheda "SIM Input" utilizzando la procedura `Clear_Sim_Input`.

In sintesi, la scheda "SIM Input" funge da luogo in cui vengono inseriti i dati di input necessari per la simulazione, e la procedura `SIM_Input` semplifica questo processo popolando la scheda con informazioni specifiche sulle attività provenienti dalla scheda "DSM".

9.8 Foglio "Probability":

La scheda "Probability" viene utilizzata per rappresentare la matrice delle probabilità associate alle attività. La procedura `Input_Probability` è responsabile per l'inserimento dei dati nella matrice delle probabilità.

9.9 Foglio “Impact”:

La scheda "Impact" viene utilizzata per rappresentare la matrice degli impatti associata alle attività. La procedura `Input_Impact` è responsabile per l'inserimento dei dati nella matrice degli impatti.

9.10 Foglio “SIM Results”:

Il codice associato alla scheda "SIM Results" si occupa principalmente della visualizzazione e analisi dei risultati ottenuti dalle simulazioni del modello. Ecco una descrizione discorsiva dei passaggi chiave:

1. **Visualizzazione dei Risultati:** I risultati delle simulazioni, in particolare le durate dei progetti, vengono copiati nella colonna B a partire dalla seconda riga della scheda "SIM Results". Questo fornisce una rappresentazione chiara e ordinata dei dati ottenuti da ciascuna simulazione.
2. **Calcolo del Foglio di Lavoro:** Dopo la copia dei risultati, il foglio di lavoro "SIM Results" viene ricalcolato. Questa operazione può essere necessaria se ci sono formule o dipendenze nelle celle che dipendono dai risultati delle simulazioni. Assicurarsi che tutti i calcoli siano aggiornati.
3. **Istogramma dei Risultati:** Viene generato un istogramma basato sulla colonna B dei risultati. L'istogramma fornisce una rappresentazione grafica della distribuzione delle durate dei progetti simulati, offrendo una visione immediata della variabilità dei risultati.
4. **Pulizia e Selezione:** Alla fine, la scheda "SIM Results" viene selezionata e la cella A1 viene evidenziata.

In sintesi, la scheda "SIM Results" agisce come un punto centrale per esplorare, interpretare e rappresentare visivamente i risultati delle simulazioni effettuate, contribuendo a una migliore comprensione delle prestazioni del modello.

9.11 Foglio “Single Run Data”:

Il codice associato alla scheda "Single Run Data" gestisce la raccolta e la visualizzazione dei dati specifici relativi a una singola esecuzione del modello. Ecco una descrizione di cosa fa la scheda "Single Run Data":

1. **Pulizia dei Dati Precedenti:** Prima di iniziare a raccogliere nuovi dati, il codice cancella eventuali dati presenti nelle celle da F4 a IV210 e da F61 a IV94 sulla scheda "Single Run Data". Questa operazione assicura che i risultati della nuova esecuzione siano visualizzati in un foglio di lavoro pulito.
2. **Inizializzazione dei Dati Iniziali:** Vengono inizializzati alcuni dati iniziali, come il numero di iterazioni (run), il numero di attività, la dimensione del passo temporale (`delta_t`), e altri parametri configurabili che possono essere necessari per la simulazione.
3. **Raccolta dei Dati per Ogni Attività:** Per ciascuna attività, vengono raccolti dati come la sequenza dell'attività, la durata campionata, lo stato di avanzamento del lavoro durante ciascun passo temporale e la rappresentazione grafica tramite un grafico di Gantt.
4. **Visualizzazione Grafica dell'Esecuzione:** Il codice genera un grafico di Gantt che rappresenta graficamente l'esecuzione delle attività nel tempo. Questo può essere utile per una rapida comprensione della pianificazione temporale e delle sovrapposizioni tra attività.
5. **Aggiornamento dei Dati Live:** Se l'esecuzione è parte di una serie di esperimenti in tempo reale (ad esempio, durante lo sviluppo o l'ottimizzazione del modello), il foglio di lavoro viene costantemente aggiornato durante l'esecuzione, mostrando come le variabili cambiano nel tempo.

In sintesi, la scheda "Single Run Data" fornisce una visione dettagliata e visuale dell'esecuzione del modello per una singola simulazione, aiutando gli utenti a comprendere meglio il comportamento del modello in termini di durate delle attività, sequenze e progresso temporale.

9.12 Aggiornamento del file Excel con Macro del Mit di Boston:

All'apertura del file dobbiamo consentire al sistema di abilitare le macro. Questa funzionalità di sicurezza permette di proteggersi da eventuali file corrotti e di evitare danni al PC. Per abilitare le macro premere il pulsante Enable Content (abilita contenuto) così come mostrato in figura 9.12.1:

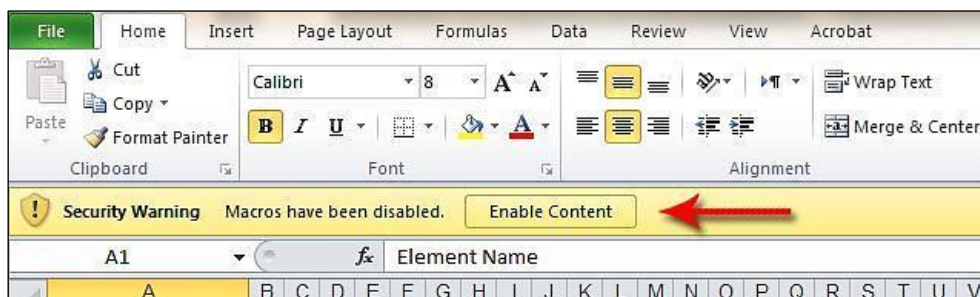


Fig. 9.12.1: schermata per l'abilitazione delle macro

Dopo aver eseguito la procedura di abilitazione delle macro si aprirà una nuova scheda nella barra degli strumenti in alto che consente di accedere alle funzionalità DSM così come mostrato in figura 9.12.2. La scheda Add-Ins (Componenti aggiuntivi) contiene Menù Commands (Comandi di Menù) e Custom Toolbars (Barre degli strumenti personalizzate):

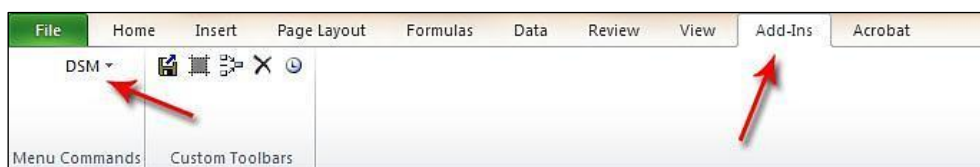


Fig. 9.12.2: verifica presenza schede abilitate

Adesso sono funzionanti tutti i fogli con le relative macro. Avvertenze importanti:

- 1) Le macro presenti nel foglio di lavoro "Activities-Parameters" utilizzano numerosi riferimenti a celle specifiche quindi è assolutamente sconsigliato modificare manualmente le posizioni relative delle tabelle.
- 2) Le formattazioni inserite sono parte del codice VBA di alcune importanti funzioni delle macro del foglio e pertanto non vanno assolutamente modificate.
- 3) Ogni cambiamento deve necessariamente avvenire tramite i pulsanti con macro.

E' possibile apportare cambiamenti manuali solo dove esplicitamente segnalato. Per qualsiasi cambiamento o aggiornamento l'utente voglia fare, nel foglio di lavoro "Activities-Parameters", l'accesso al pannello di controllo per le macro e la programmazione VBA si attiva con la combinazione di tasti ALT + F11 e va selezionato il modulo chiamato moduloDG.

Prima di utilizzare il foglio di lavoro è necessario che le macro siano attive (così come indicato nella sezione precedente) e che siano aggiornati gli strumenti di analisi VBA che consentono a tutto il file di funzionare correttamente. Seguire queste istruzioni: andare su file > altro....> opzioni come mostrato in figura 9.12.3.

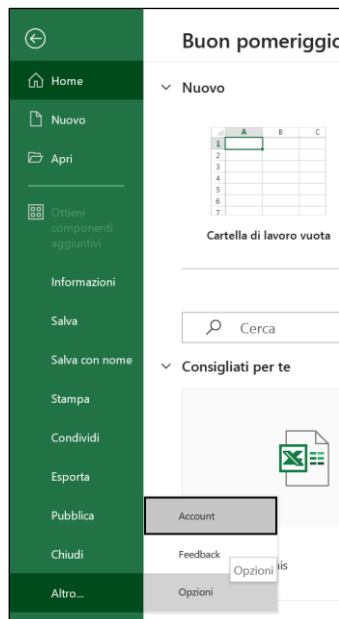


Fig 9.12.3: schermata della pagina Home

dalla pagina Opzioni di Excel selezionare > componenti aggiuntivi: viene offerta la vista del box utente in figura 9.12.4.

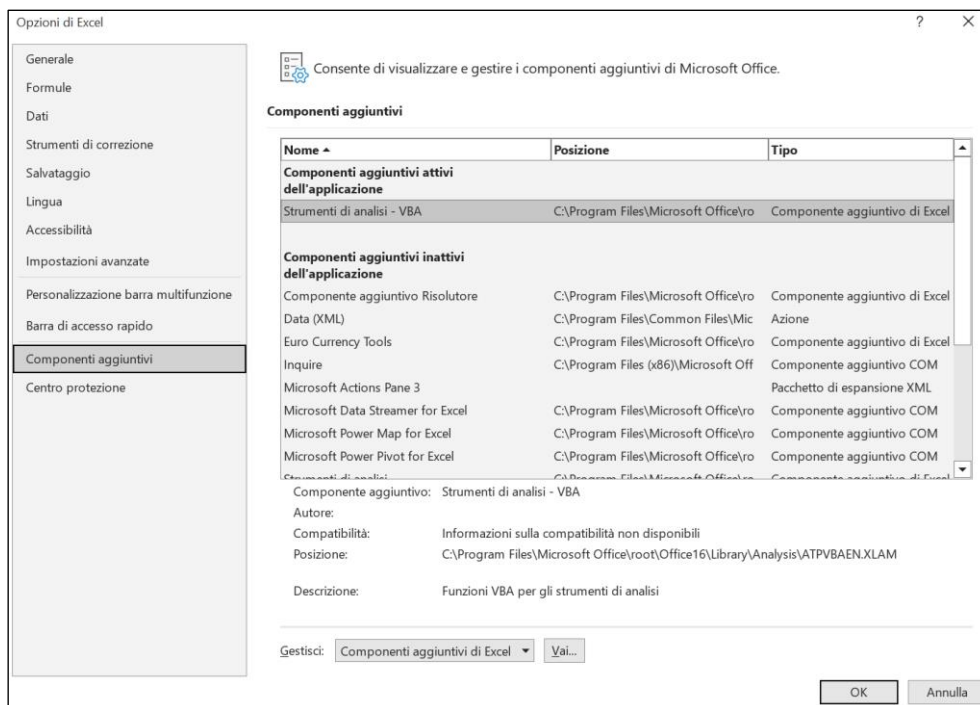


Fig 9.12.4: finestra di dialogo componenti aggiuntivi

da Componenti aggiuntivi inattivi dell'applicazione selezionare > strumenti di analisi – VBA e premere Vai...

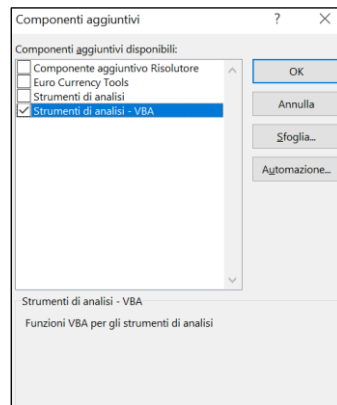


Fig.9.12.5: finestra di dialogo componenti aggiuntivi disponibili

dalla finestra componenti aggiuntivi selezionare con una spunta > strumenti di analisi – VBA e premere OK: il box viene mostrato in figura 9.12.5. Adesso la funzionalità ATPVBAEN.XLA!Histogram, in un formato che ormai è divenuto obsoleto, è stata sostituita con ATPVBAEN.XLAM!Histogram con estensione XLAM: la pagina Sim Input funziona correttamente. Verificarlo se possibile sul modulo specifico della programmazione VBA che è il modulo 9: in figura 9.12.6 si mostra la schermata di riferimento.

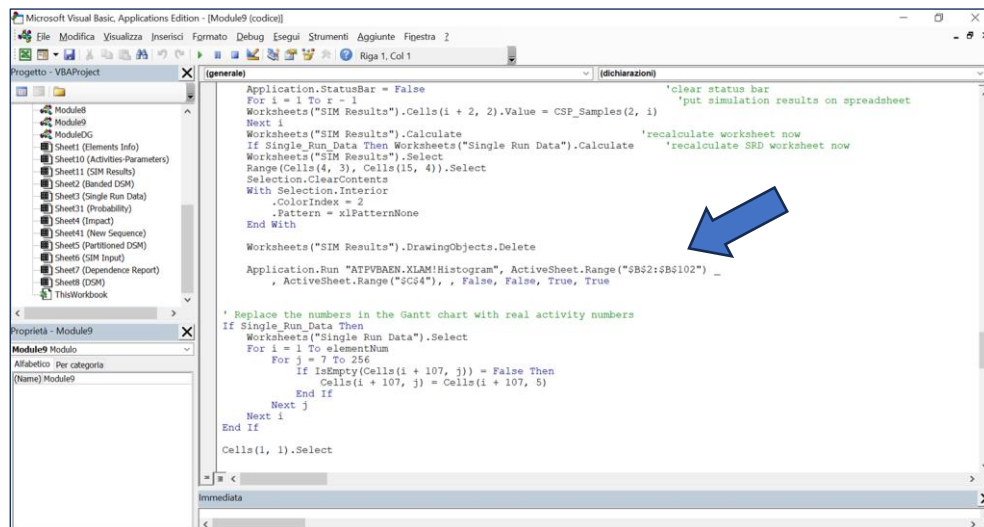


Fig. 9.12.6: schermata di visualizzazione del codice VBA corretto

9.13 Istruzioni di utilizzo della pagina “Activities-Parameters”:

Quelli elencati di seguito sono i 12 fogli di lavoro presenti nel file del Mit di Boston e quello evidenziato è il foglio che ho implementato: si osservi figura 9.13.1 per vedere la sua posizione.



Fig 9.13.1: elenco dei fogli con evidenziato l’inserimento del foglio Activities-Parameters

Si raccomanda di seguire il più fedelmente possibile le istruzioni di seguito riportate poiché un uso non corretto renderà le macro inutilizzabili! Dopo aver cliccato sul foglio di lavoro “Activities-Parameters” in figura 9.13.2 è mostrato quello che dovrebbe apparire:

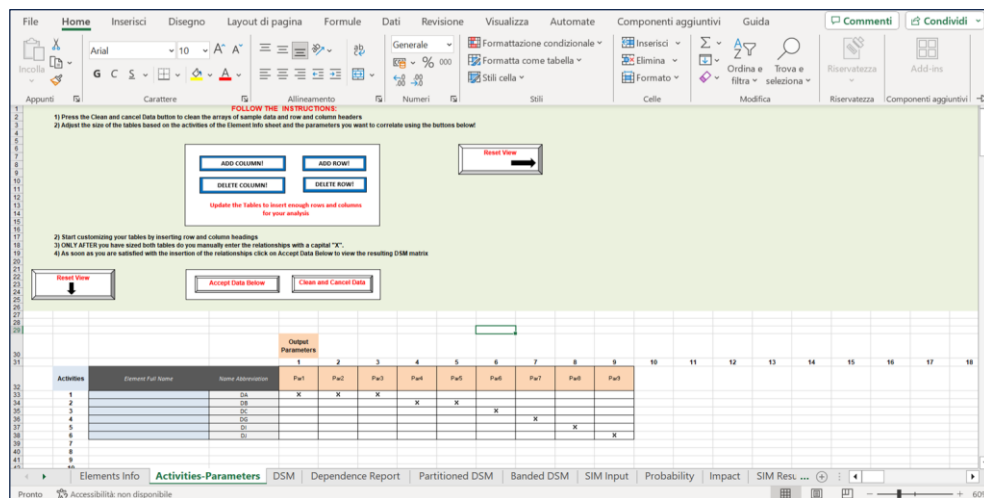


Fig. 9.13.2: schermata iniziale all’apertura del foglio di lavoro

Questo foglio di lavoro produce una matrice DSM di dimensioni massime 50X50 a partire da due matrici DMM di dimensioni massime 50X20 dove 20 è il numero massimo di parametri implementabili: è importante tenere a mente le attuali limitazioni del programma e regolare lo studio di conseguenza. Impostare lo zoom al 13% per evidenziare le aree di lavoro: la zona in alto con banda colorata comprende le istruzioni e i pulsanti collegati a funzionalità macro. Ci sono tre zone: la zona della tabella Attività Parametri di Output, la zona della tabella Attività Parametri di Input e la zona della matrice DSM. Per una migliore lavorabilità all’interno del foglio impostare lo zoom al 60% posizionare il cursore sulla cella L43 dove è scritto “NON CANCELLARE” poi dal menu Visualizza andare nella zona finestra e selezionare dividi. Adesso il foglio è diviso in quattro sezioni che possono scorrere utilizzando sia le barre laterali che la rotella del mouse dopo aver cliccato una cella dentro l’area. Spostare manualmente la divisione orizzontale fino a trovarsi tra la riga 25 e la riga 26. La divisione verticale deve essere posizionata tra la colonna K e la colonna L. Cliccare una cella del riquadro in basso a sinistra e poi fare clic con il mouse sul pulsante Reset View con la freccetta verso il basso. Fare la stessa cosa sul riquadro in alto a destra: cliccare con il mouse su una cella del riquadro e fare clic sul pulsante Reset view con la freccetta verso destra come indicato in figura.

Adesso la pagina è impostata correttamente per l'inserimento dei dati e dovrebbe apparire come in figura 9.13.3:

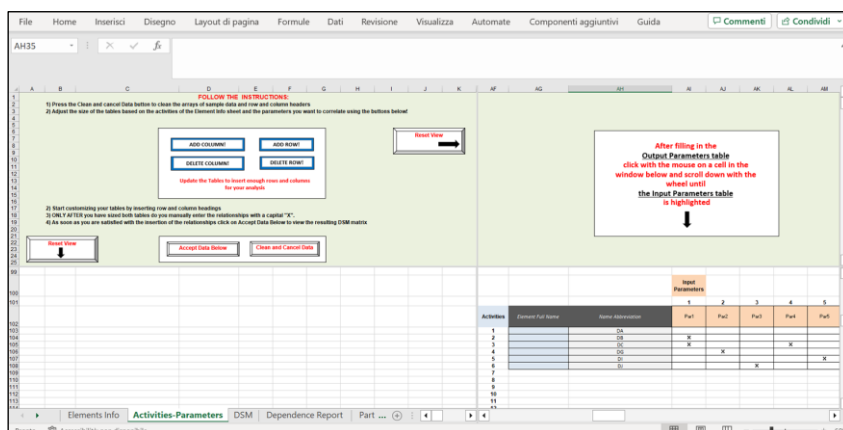


Fig.9.13.3: schermata di configurazione iniziale

L'inserimento dati avviene prima popolando la tabella che compare nel riquadro in basso a sinistra ma ancor prima di dedicarci all'inserimento delle relazioni dobbiamo dimensionare correttamente, secondo le nostre esigenze, la tabella agendo sui pulsanti del riquadro centrale: è possibile aggiungere o cancellare una riga oppure aggiungere o cancellare una colonna. Non è possibile cancellare tutte le righe o tutte le colonne in quanto se ci proviamo viene segnalato un messaggio di errore dal sistema. Premere il pulsante Clean and Cancel Data per liberare le tabelle dai dati di esempio. La formattazione va lasciata così come è e non va cambiata manualmente in quanto anche lei contribuisce al funzionamento del codice VBA. Mentre cambiamo le dimensioni della tabella si modifica contemporaneamente anche la seconda tabella che ancora non è visibile e si modifica la tabella della matrice DSM. Quindi, riepilogando, le modifiche vanno fatte solo alla prima tabella e poi si trasmettono in automatico al resto delle tabelle presenti nel foglio. Dopo aver dimensionato la tabella si può personalizzare il nostro studio inserendo le intestazioni di riga e di colonna e le abbreviazioni, se necessarie, degli elementi di riga che costituiscono le attività. Si può adesso inserire le relazioni attraverso l'immissione di "X" maiuscole. La prima tabella è completata. Ci si mette con il cursore sul riquadro in basso a destra e si agisce sulla rotella del mouse o sulla barra laterale per scorrere verso il basso fino ad evidenziare la seconda tabella che troviamo già dimensionata e pronta per essere solo personalizzata con l'inserimento delle informazioni necessarie. Questa modalità è scritta sul riquadro in alto a destra che riporta le istruzioni da seguire. E' importante ribadire che le relazioni vanno descritte con l'inserimento di "X" maiuscole! In figura 9.13.4 si vede la schermata con la seconda tabella pronta:

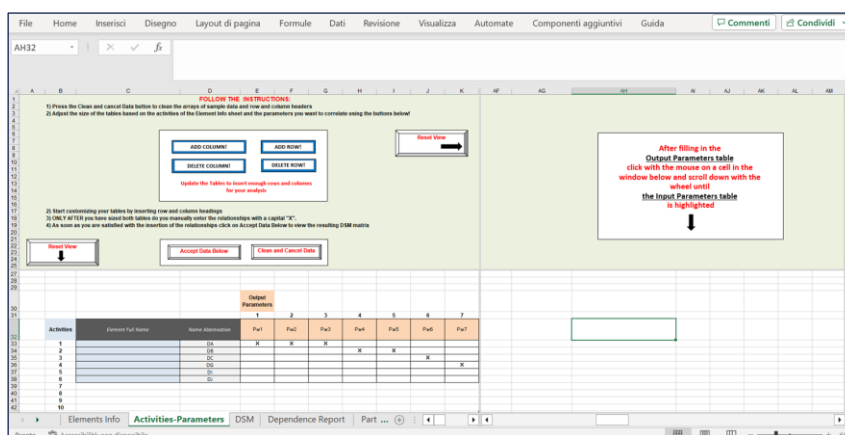


Fig. 9.13.4: schermata con la tabella degli Input

Siamo pronti per accettare i dati inseriti e generare la matrice DSM che sarà disponibile premendo il **pulsante Accept Data Below**. La figura 9.13.5 mostra quello che apparirà all'utente:

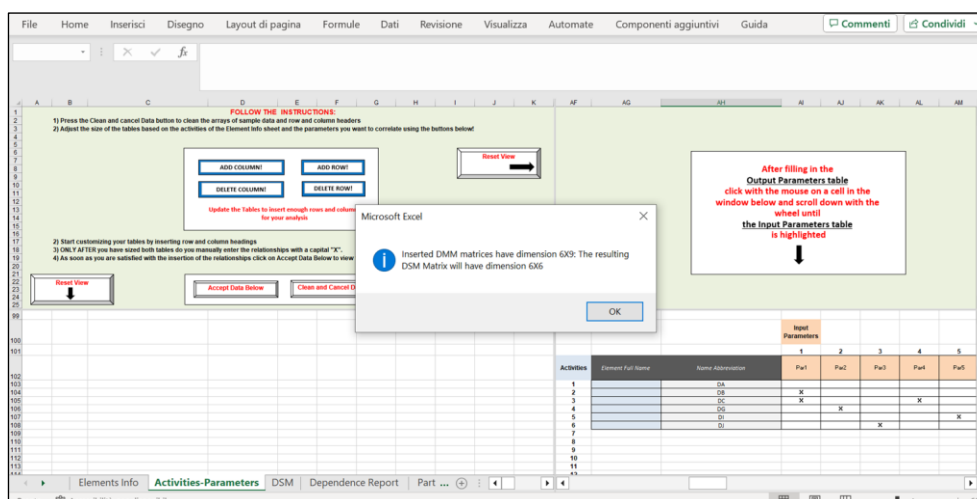


Fig. 9.13.5: comparsa del box con l'avviso delle dimensioni delle matrici

Il box riporta l'informazione delle dimensioni delle tabelle dei dati appena riempite e l'indicazione della dimensione della matrice DSM che verrà generata. Appena premiamo il pulsante ok viene mostrato nel riquadro in basso a destra la matrice DSM con l'inserimento delle relazioni tra le attività. La relazione viene evidenziata dalla presenza del numero 1. in figura 9.13.6 si mostra ciò che apparirà a schermo:

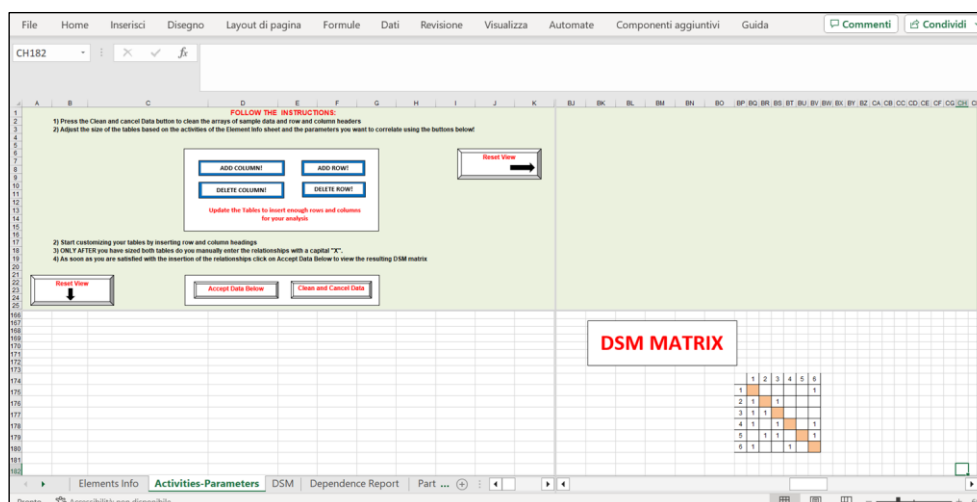


Fig. 9.13.6: comparsa della matrice DSM elaborata

A questo punto sarà possibile copiare e incollare nel foglio di lavoro DSM il risultato ottenuto e dopo avere aggiornato i dati, procedere con il nostro studio del processo produttivo. Qualsiasi cambiamento vogliamo apportare non dobbiamo agire sulla matrice manualmente ma premere il **pulsante Clean and Cancel Data** e ricominciare da capo con le modifiche sulla tabella di Output. Questo pulsante cancella la matrice DSM e cancella i dati inseriti nelle due tabelle. Faremo clic con il mouse su una cella del riquadro in basso a sinistra e premeremo il pulsante Reset View con la freccetta verso il basso per ritornare a visualizzare la tabella di Output poi faremo clic su una cella del riquadro in alto a destra e premeremo il pulsante Reset View con la freccetta verso destra per ritornare a visualizzare il riquadro con le istruzioni. Saremo pronti ora a ricominciare con le operazioni di modifica. Ovviamente se si è più a nostro agio con il foglio di lavoro a schermo intero si può navigare nella pagina tramite lo zoom e le barre laterali di spostamento e per questo lascio indicate le coordinate

delle celle che identificano le tabelle per poterle ritrovare con facilità nei modi che riterrete opportuno. Fare riferimento alla sezione Celle di riferimento importanti.

9.14 Celle di riferimento importanti

Tabella Attività-Parametri di Output, come si vede nell'esempio di figura 9.14.1: riferimento di cella **B30**

			Output Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA	x	x	x						
2		DB				x	x				
3		DC						x			
4		DG							x		
5		DI								x	
6		DJ									x

Fig. 9.14.1: tabella Attività-Parametri Output di esempio

Tabella Attività-Parametri di Input, come si vede nell'esempio di figura 9.14.2: riferimento di cella **AF100**

			Input Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA									x
2		DB	x					x			
3		DC	x			x					
4		DG		x				x			x
5		DI					x	x			x
6		DJ			x				x		

Fig. 9.14.2: tabella Attività-Parametri Input di esempio

Matrice DSM, come si evince dalla figura 9.14.3: riferimento di cella **BP174**

	DA	DB	DC	DG	DI	DJ
DA						1
DB	1		1			
DC	1	1				
DG	1		1			1
DI		1	1			1
DJ	1			1		

Fig. 9.14.3: matrice DSM di esempio

9.15 Spiegazione dell'algoritmo per la formazione della matrice DSM:

Calcoliamo la prima colonna della matrice DSM di modo che il ragionamento di costruzione apparirà chiaro. Con questo tipo di costruzione la convenzione che adottiamo sulla matrice DSM è la convenzione IR (Input in Row) che è la medesima adottata nei calcoli affrontati nel foglio di lavoro Excel. Il metodo prevede di scorrere tutte le relazioni sulla prima riga della tabella Attività-Parametri di Output (prima colonna della matrice DSM) e trovare tutte le relazioni che si corrispondono tramite i segni "X" nella seconda tabella così come mostrato in figura 9.15.1.

Passaggio 1:

			Output Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA	X	X	X						
2		DB				X	X				
3		DC						X			
4		DG							X		
5		DI								X	
6		DJ									X

			Input Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA									X
2		DB	X					X			
3		DC		X		X					X
4		DG					X	X			
5		DI					X	X			
6		DJ			X				X		

Fig. 9.15.1: elaborazione del passaggio 1

L'inserimento del dato avviene partendo dalla prima colonna DA e pone la relazione sulla riga DB attraverso l'inserimento del numero 1, così come mostrato in figura 9.15.2. L'elemento è stato posizionato correttamente nella matrice DSM.

	DA	DB	DC	DG	DI	DJ
DA						
DB	1					
DC						
DG						
DI						
DJ						

Fig. 9.15.2: DSM relativo al passaggio 1

Passaggio 2:

La prima colonna della seconda tabella presenta un'altra relazione che va considerata così come mostrato in figura 9.15.3.

			Output Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA	x	X	X						
2		DB				X	X				
3		DC						X			
4		DG							X		
5		DI								X	
6		DJ									X

			Input Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA									X
2		DB				X		X			
3		DC		X							
4		DG					X	X			X
5		DI			X				X		
6		DJ									

Fig. 9.15.3: elaborazione del passaggio 2

L'inserimento del dato nella matrice DSM avviene partendo sempre dalla prima colonna DA e pone la relazione 1 sulla riga DC, così come si vede in figura 9.15.4.

	DA	DB	DC	DG	DI	DJ
DA						
DB	1					
DC	1					
DG						
DI						
DJ						

Fig. 9.15.4: DSM relativo al passaggio 2

Passaggio 3:
 La seconda colonna della prima tabella presenta un segno X a cui corrisponde nella seconda tabella un segno X che riporta alla riga 4, come si evince da figura 9.15.5.

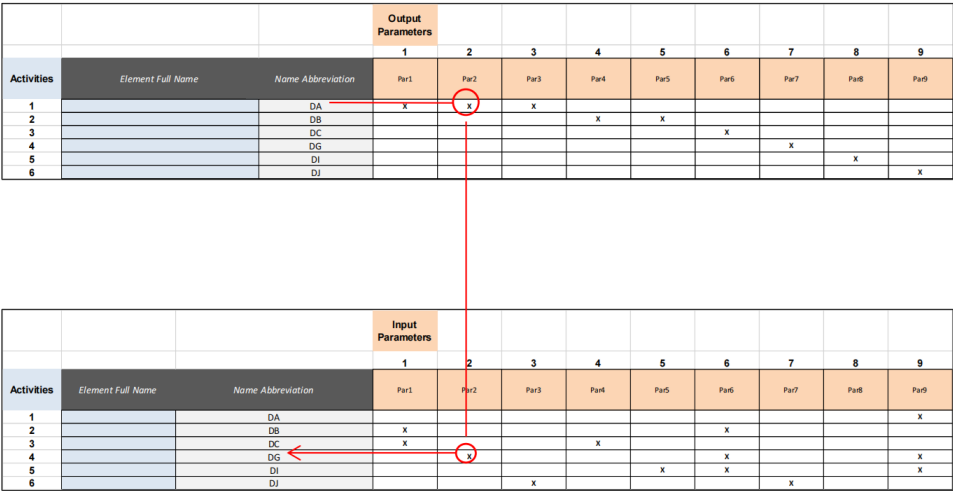


Fig. 9.15.5: elaborazione del passaggio 3

L’inserimento del dato avviene partendo ancora dalla prima colonna DA e pone la relazione 1 sulla riga DG, come mostrato in figura 9.15.6.

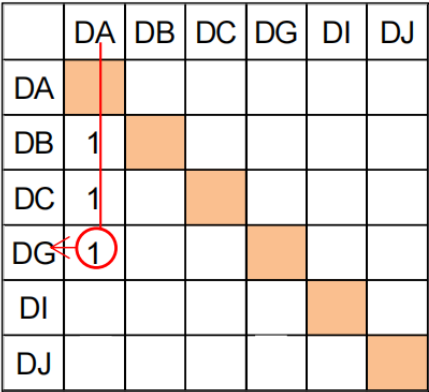


Fig. 9.15.6: DSM relativo al passaggio 3

Passaggio 4:

La terza colonna della prima tabella presenta un segno X a cui corrisponde nella seconda tabella un segno X che fa riferimento alla riga 6, come si vede in figura 9.15.7.

			Output Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA	X	X	X						
2		DB				X	X				
3		DC						X			
4		DG							X		
5		DI								X	
6		DJ									X

			Input Parameters								
			1	2	3	4	5	6	7	8	9
Activities	Element Full Name	Name Abbreviation	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8	Par9
1		DA									X
2		DB	X					X			
3		DC	X			X					
4		DG		X				X			X
5		DI					X	X			X
6		DJ			X				X		

Fig. 9.15.7: elaborazione del passaggio 4

Inseriamo l'ultima relazione della colonna DA posizionando 1 sulla riga DJ, come si mostra in figura 9.15.8.

	DA	DB	DC	DG	DI	DJ
DA						
DB	1					
DC	1					
DG	1					
DI						
DJ	1					

Fig. 9.15.8: DSM relativo al passaggio 4

La prima colonna è stata completata e si passa alla seconda colonna, cioè consideriamo la riga DB della tabella attività parametri di Output e procediamo in modo del tutto analogo a prima.

Alla fine del processo la matrice DSM viene popolata di tutte le relazioni e quello mostrato in figura 9.15.9 è il risultato finale.

	DA	DB	DC	DG	DI	DJ
DA						1
DB	1		1			
DC	1	1				
DG	1		1			1
DI		1	1			1
DJ	1			1		

Fig. 9.15.9: DSM finale dopo l'elaborazione

10 Sviluppo del Plugin Obsidian “Adjacency Matrix Exporter”:



In questo capitolo 10 si affronta il lavoro fatto con il software Obsidian.md. Nei primi paragrafi viene presentato il software e descritto nelle sue parti principali, si illustrano poi le funzionalità più importanti e le motivazioni che hanno portato alla sua scelta. Sono state approfondite alcune funzionalità importanti per lo sviluppo del lavoro successivo come la funzionalità Canvas. Viene poi descritto nel dettaglio il codice programmato per il metodo `ExportCsvAbsolute()` in linguaggio Typescript, a titolo di esempio, dove compare l'implementazione della matrice di adiacenza dei dati nella modalità Absolute. Oltre al corpo centrale della programmazione del codice, sono state utilizzate anche istruzioni HTML e CSS per migliorare l'aspetto grafico delle varie finestre di interazione utente producendo due file esterni al codice principale. E' importante sottolineare che lo strumento è il primo nel suo genere all'interno di Obsidian e che probabilmente sarà la base per ulteriori sviluppi futuri di implementazioni della teoria trattata. Il plugin “Adjacency matrix Exporter” è entrato a far parte della Community dei plugin scaricabili da Obsidian in data 6 Dicembre 2023 attraverso un processo di revisione da parte di due professionisti programmatori durato un mese e che collaborano con la piattaforma di Obsidian. Attualmente il plugin nominato fa parte di una lista messa a disposizione dalla Community e liberamente scaricabile dagli appassionati di tutto il mondo.

10.1 Il software Obsidian.md

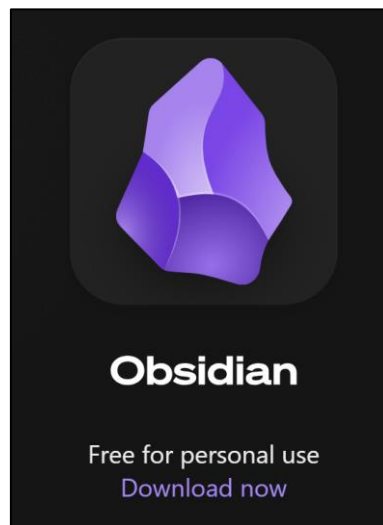


Fig. 10.1.1: logo della schermata principale del software

Obsidian.md, il cui logo è mostrato in figura 10.1.1, è un software open source lanciato nel 2020, sviluppato da due giovani ragazzi di origine cinese Shida Li e Erica Xu (università di Waterloo – Canada) durante il periodo della pandemia Covid, diventato rapidamente popolare tra coloro che cercano un'applicazione per la gestione delle note basata sul Markdown con funzionalità di collegamento e organizzazione avanzato. Obsidian è un software per la gestione delle note e la creazione di collegamenti tra di esse. Alcune caratteristiche peculiari sono le seguenti:

- 1. Markdown-Based:** Obsidian.md utilizza la sintassi Markdown per formattare le note, rendendo la scrittura e la modifica facile e intuitiva.
- 2. Grafo delle connessioni:** attraverso il software è possibile creare collegamenti tra le note in modo da visualizzare un grafo delle connessioni, aiutando a organizzare le idee e vedere come i concetti sono correlati.
- 3. Editor Markdown:** offre un editor Markdown con anteprima in tempo reale che permette di vedere come apparirà il testo formattato mentre scrivi.
- 4. Ricerca Potente:** Obsidian.md dispone di un potente motore di ricerca che aiuta a trovare rapidamente le note anche tra migliaia di documenti.
- 5. Plugin e Personalizzazione:** si tratta di un software altamente personalizzabile grazie all'ecosistema di plugin che permette di estendere le funzionalità.
- 6. Open Source:** Obsidian è open source, il che significa che puoi accedere al codice sorgente e modificarlo secondo le tue esigenze.
- 7. Sincronizzazione:** è possibile sincronizzare le note attraverso servizi come Dropbox o altro per accedervi da diversi dispositivi.
- 8. Versioning e Backup:** Obsidian offre il supporto per il versioning delle note ed è possibile eseguire il backup dei dati in modo da non perdere mai informazioni importanti. Il versioning delle note è un sistema che tiene traccia delle diverse versioni di una nota o di un documento nel tempo. Ogni volta che si apportano modifiche, il sistema registra e conserva una copia di quella nota nello stato precedente: si ha quindi l'opportunità di ripristinare versioni precedenti, tracciare le modifiche nel tempo ed avere una maggiore sicurezza dei dati che se si corrompono possono essere recuperati in una versione precedente al danneggiamento.

10.2 Scopo e descrizione del funzionamento del plugin “Adjacency Matrix Exporter”

Exporter”

Questo Plugin mette in luce la corrispondenza tra la teoria dei Grafi e la struttura del software Obsidian.md che gestisce le note e i links che li collegano costruendo una rete di informazioni a formare quello che viene definito un secondo cervello (Second Brain). Dalla teoria dei grafi deriviamo il concetto di Matrice di adiacenza con pesi, descritto in precedenza, e tale concetto lo ricaviamo in modo naturale all'interno di Obsidian se identifichiamo i nodi con le note, gli edge o archi con i links tra le note e definiamo peso il numero totale di collegamenti che vanno dalla nota i alla nota j. Lo script creato, il cui aspetto all'apertura è mostrato in figura 10.2.1, ha la finalità di esportare in formato CSV la matrice di adiacenza con pesi che si può costruire collegando tra loro le note del Vault attraverso dei links ed è possibile farlo secondo due modalità che sono state chiamate: Absolute e Normalized.

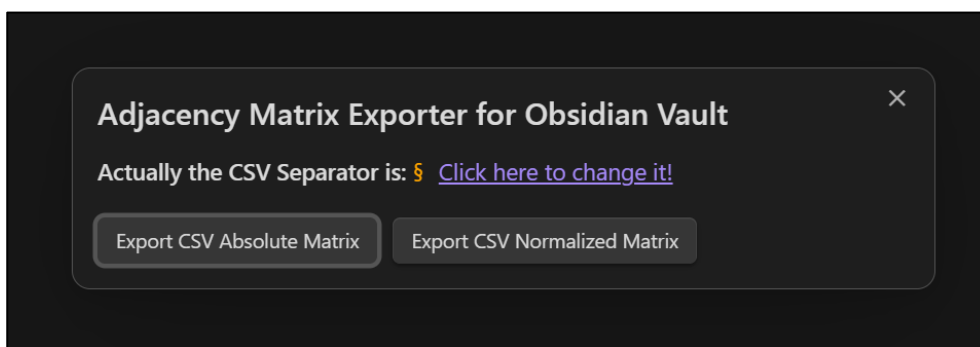


Fig 10.2.1: Aspetto della finestra interazione utente all'apertura del Plugin

Premendo con un click del mouse il pulsante della matrice, da esportare nella versione desiderata, viene generato un messaggio in alto a destra di avviso che il file è stato generato come si vede in figura 10.2.2. Nella matrice di adiacenza Absolute, siccome il collegamento tra una nota e l'altra può non essere unico, se tra la nota i e la nota j abbiamo p collegamenti (peso dei links) nella casella relativa all'edge compare il numero p. Il caso è esemplificato in figura 10.2.4.

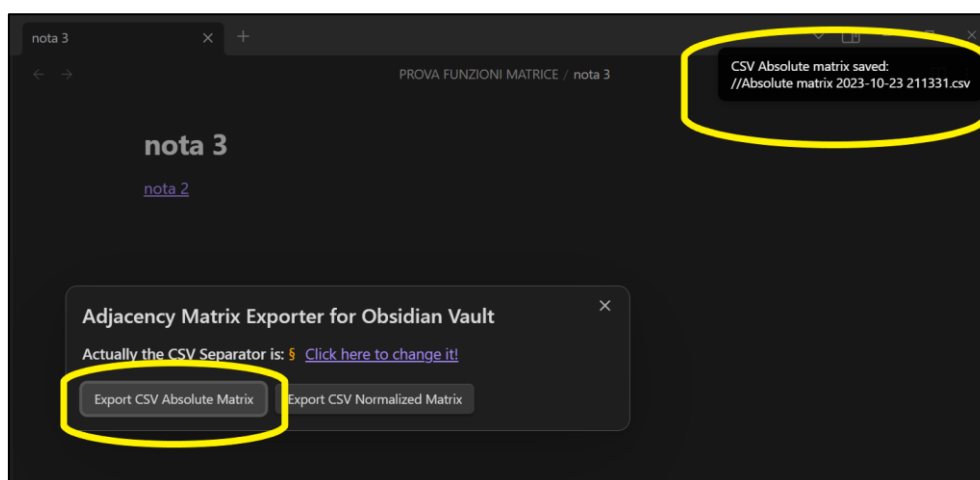


Fig. 10.2.2: generazione in alto a destra dell'avviso di salvataggio del file

Il file, che di default viene posizionato all'interno della cartella Obsidian Vault, come mostra la figura 10.2.3, può essere aperto tramite Excel o LibreOffice o altro software dedicato: si evidenzia il nome della modalità di esportazione, la data nel formato AAAA-MM-GG e indicazione dell'ora di creazione.

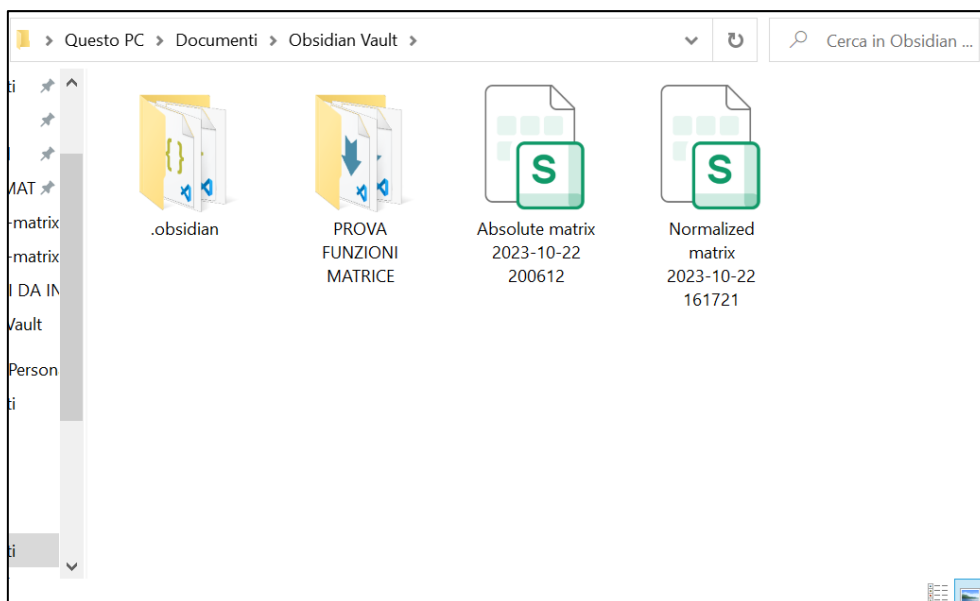


Fig 10.2.3: i file generati vengono salvati nella cartella Obsidian Vault

	A	B	C	D
1		nota 1	nota 2	nota 3
2	nota 1	0	6	5
3	nota 2	2	0	0
4	nota 3	0	1	0
5				

Fig. 10.2.4: la matrice di adiacenza nella versione Absolute

Nella matrice di adiacenza, versione Normalized, il peso viene normalizzato attraverso la divisione di p per il numero totale delle parole contenuto nella nota origine. Il file appare come nella figura 10.2.5: è da notare come i numeri decimali compaiano con la virgola e non con il punto. Questo scambio, necessario per far riconoscere al programma destinatario il numero decimale in quanto tale, è stato ottenuto a livello di codice tramite la funzione **replace** con cui abbiamo sostituito il punto con la virgola. Senza questo artificio il numero comparirebbe con il punto e il programma destinatario lo interpreterebbe come stringa.

	A	B	C	D
1		nota 1	nota 2	nota 3
2	nota 1	0	0,272727272727273	0,227272727272727
3	nota 2	0,5	0	0
4	nota 3	0	0,5	0
5				
6				

Fig. 10.2.5: la matrice di adiacenza versione Normalized

Il file CSV che viene prodotto ha un separatore di default (Csv Separator) che può essere cambiato a seconda delle esigenze dell'utente tramite click del link colorato come si vede in figura 10.2.6 e interazione con la finestra di dialogo che si mostra all'utente nella figura 10.2.7.

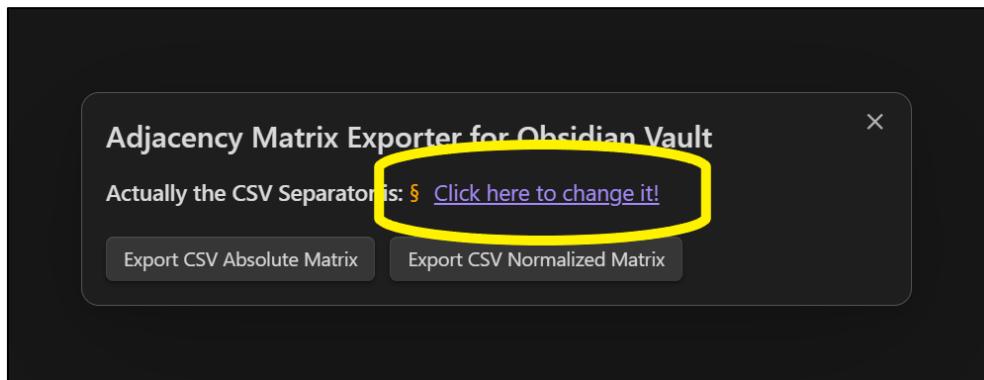


Fig. 10.2.6: link che consente il cambio di separatore

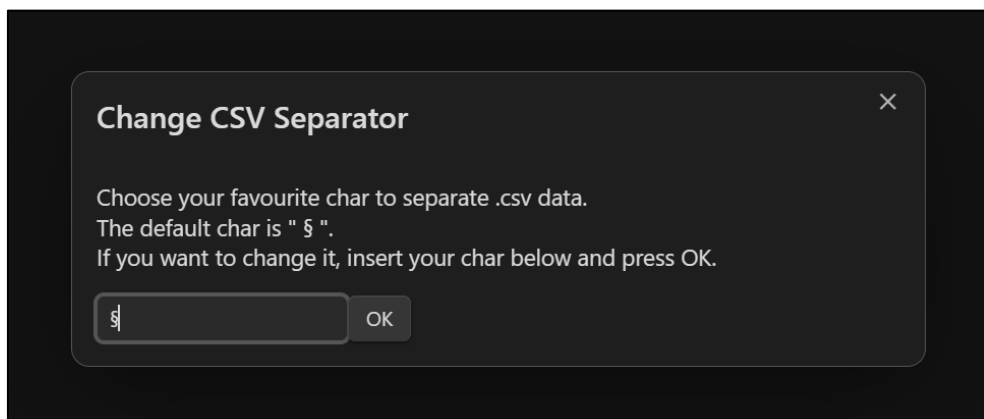


Fig. 10.2.7: box interazione utente per il cambio di separatore del file Csv

Dalle opzioni accessibili tramite link, come si vede in figura 10.2.8, è possibile impostare un separatore in modo permanente che diventerà quello di default e sarà possibile cambiare la cartella su cui vengono salvate le matrici di adiacenza che è per default la cartella Obsidian Vault.

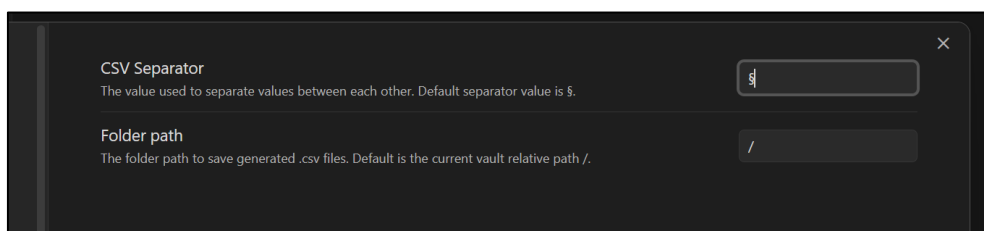


Fig. 10.2.8: finestra di dialogo delle opzioni del Plugin

10.3 Spiegazione della struttura del plugin

La scrittura del codice di questo plugin cerca di seguire un principio molto importante in programmazione e cioè quello della modularità. Questo comporta la suddivisione di un sistema complesso in parti più piccole e gestibili chiamate “moduli” o “componenti”. Tali moduli sono

progettati in modo da svolgere funzioni specifiche e ben definite. Un esempio comune di modularità è l'uso di funzioni in un linguaggio di programmazione. Ogni funzione svolge una specifica attività e può essere chiamata da altre parti del codice quando necessario. Il linguaggio utilizzato per la stesura di questo codice è Typescript: un superset di JavaScript, che aggiunge tipi, classi, interfacce e moduli opzionali al JavaScript tradizionale. Typescript è in sostanza una estensione di JavaScript. La struttura fondamentale del codice di programmazione del Plugin “Adjacency Matrix Exporter” si compone di 2 Classi e di 6 funzioni principali di cui 2 sono funzioni asincrone.

Una classe è una struttura fondamentale per la programmazione ad oggetti: questi oggetti possono avere proprietà (variabili) e metodi (funzioni). Le classi in Typescript forniscono un modo efficace per organizzare e strutturare il codice consentendo la creazione di oggetti con comportamenti e attributi ben definiti. In questo codice ho fatto uso di funzioni così dette regolari (o sincrone) e funzioni asincrone: la differenza principale sta nella diversa gestione delle operazioni che richiedono tempo, come l'input/output o le richieste di rete.

Le funzioni sincrone eseguono le istruzioni in modo sequenziale, una dopo l'altra; bloccano l'esecuzione del programma fino a quando l'operazione in corso non è completata; sono adatte per operazioni rapide o calcoli che non richiedono attesa.

Le funzioni asincrone consentono di avviare operazioni in background senza bloccare l'esecuzione del programma; utilizzano le callback, le Promise o l'async/await per gestire l'asincronia; sono adatte per operazioni che richiedono tempo, come richieste di rete o lettura/scrittura di file. Le funzioni asincrone sono utili quando si vuole evitare il blocco dell'interfaccia utente in applicazioni web, o quando si devono gestire molteplici operazioni parallele.

10.3.1 Il metodo exportCsvAbsolute()

```
// export Absolute matrix scores as CSV
exportCsvAbsolute() {
  const app = this.app;
  const settings = this.settings;
  const files = this.files;
  console.log("Save CSV!");
  const symbol = this.settings.csvSeparator;
  const folderPath = settings.folderPath === "" ? "/" : settings.folderPath;
  const now = window.moment().format("YYYY-MM-DD HHmmss");
  const filePath = `${folderPath}/Absolute matrix ${now}.csv`;
  const size = files.length;
  let data = "";
  // print CSV header
  for (let j = size - 1; j >= 0; j--) {
    data += symbol + files[j].basename;
  }
  data += "\n";
  for (let i = size - 1; i >= 0; i--) {
    data += files[i].basename + symbol;
    for (let j = size - 1; j >= 0; j--) {
      const n_links = app.metadataCache.resolvedLinks[files[i].path][files[j].path]
      data += isNaN(n_links) ? 0 : n_links;
      if (j > 0)
        data += symbol;
    }
    data += "\n";
  }
  console.log("Separator used is, " + symbol);
  console.log(data);
  app.vault.create(filePath, data);
  new Notice(`CSV Absolute matrix saved:\n${filePath}`);
}
```

Questo metodo è progettato per generare un file CSV contenente una matrice di dati. Utilizza alcune variabili per configurare il percorso del file, il separatore CSV e i dati dei file. Successivamente genera l'intestazione del CSV e riempie le righe dei dati utilizzando i collegamenti risolti tra i file. Infine, crea il file CSV e mostra una notifica all'utente. La parte più importante di questo codice è quella che

calcola i valori n_links e che permette di conoscere il peso relativo al numero di collegamenti tra le note. Il calcolo è possibile tramite l'API Obsidian: `app.metadatacache.resolvedlinks`. Data è l'array matriciale che si occupa di registrare al suo interno i risultati dei cicli for che si susseguono sull'array files. Il primo ciclo for, che scorre sul parametro j fino a $size-1$, imposta un conteggio all'indietro dell'array files producendo la prima riga della matrice con le intestazioni delle note: questo conteggio all'indietro è necessario in quanto l'API di Obsidian registra su files i nuovi dati inserendoli sempre in prima posizione e scorrendo i precedenti verso destra. Senza questo accorgimento il risultato sarebbe che la nota 1, inserita per prima, comparirebbe nella matrice in ultima posizione. Si passa alla riga successiva dove viene impostato un ulteriore ciclo for per il conteggio del totale dei collegamenti tra le note tramite il valore n_links . Man mano che i parametri di conteggio scorrono tra un valore e l'altro viene inserito il separatore csv impostato. Al completamento della matrice è possibile visualizzare il risultato sulla console di Obsidian richiamabile premendo la combinazione di tasti CTRL+MAIUSC+I. Viene visualizzato il separatore usato, la matrice dei dati con i valori separati dal simbolo csv.

10.4 La funzionalità canvas di Obsidian.md

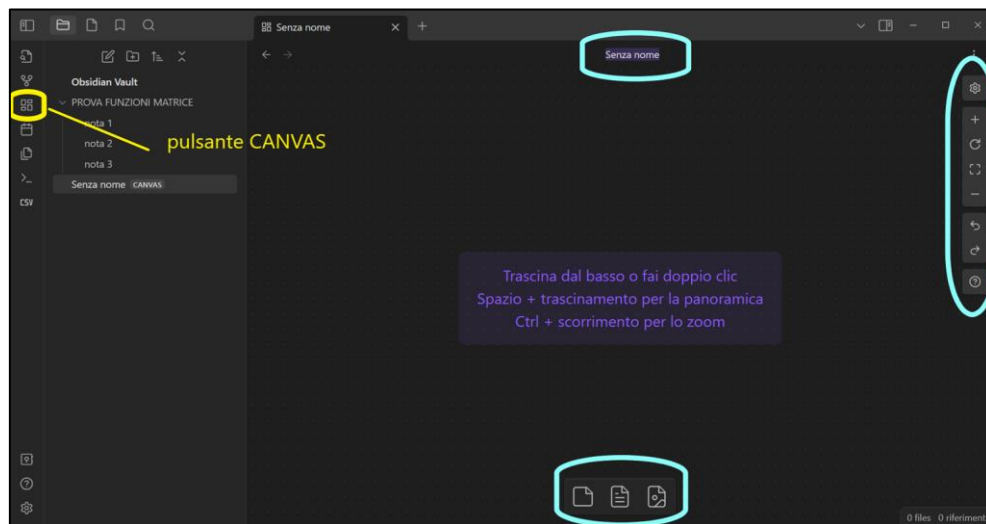


Fig. 10.4.1: schermata iniziale

La funzionalità CANVAS viene attivata premendo il pulsante relativo, come segnato in figura, sulla barra delle applicazioni a sinistra. Ho deciso di approfondire questo aspetto del programma Obsidian perché utilizzerò lo strumento per affrontare la costruzione della matrice di adiacenza del caso studio: in questo modo avrò la possibilità di avere una descrizione visuale e chiara del procedimento adottato e di poterla spiegare facilmente. La lavagna di lavoro si attiva cliccando sull'icona evidenziata in figura 10.4.1 e mostrata sotto in figura 10.4.2.



Fig. 10.4.2: pulsante della funzione Canvas

10.4.1 Aggiunta del titolo

Si apre l'interfaccia con la lavagna pronta per iniziare il lavoro: in alto, come evidenziato in figura 5.10, si può inserire, facendo click con il tasto sinistro del mouse sulla riga relativa, il titolo della lavagna. Viene creato a questo punto un file .canvas sul Vault al cui interno verranno inserite tutte le

nostre interazioni. Nell'interfaccia notiamo subito la comparsa di una griglia, tre icone cliccabili in basso e delle icone cliccabili in alto a destra come segnato in figura con il colore azzurro. Partiamo con la descrizione delle tre icone in basso evidenziate in figura 10.4.1.1:

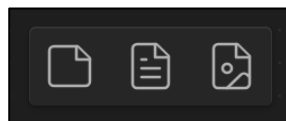


Fig. 10.4.1.1: icone utilizzabili

10.4.2 Aggiunta di una annotazione

la prima icona a sinistra ci consente di creare un'annotazione. Se ci andiamo sopra con il mouse e tenendo premuto il tasto sinistro trasciniamo sulla lavagna e poi rilasciamo, si crea il box dell'annotazione. E' possibile scrivere al suo interno ciò che vogliamo e possiede una serie di funzionalità incorporate visibili nelle icone che compaiono nella parte alta del box: rimuovi, imposta colore, zoom selezione (che può essere anche effettuato da tastiera tramite la combinazione SHIFT + 2), modifica. Cliccando al di fuori del box con il tasto sinistro viene generata l'annotazione con la scritta al suo interno ed il riquadro che la identifica che diventa da colorato con il bordo viola a colore di default grigio. Il box è totalmente modificabile nelle dimensioni tramite l'utilizzo del mouse: avvicinando il puntatore al bordo la freccetta diventa una doppia freccia e ci indica la possibilità di modifica. La modifica si può fare in orizzontale, verticale e diagonale. Si nota che percorrendo con il puntatore il bordo del box arrivati in prossimità del centro di ogni lato compare un pallino viola e il puntatore diviene una manina con l'indice. Tenendo premuto il tasto sinistro e trascinando si evidenzia una freccia di collegamento che darà la possibilità, una volta rilasciato il tasto, di creare una nota collegata come una nota dal Vault oppure come una semplice annotazione (questa possibilità è data tramite il menù a tendina che compare al rilascio del tasto sinistro del mouse).

10.4.3 Aggiunta di una nota dal Vault

La seconda icona in basso consente di aggiungere alla lavagna una nota del Vault che sarà completamente modificabile sia come box, all'interno della lavagna, che come contenuto. Anche questa nota, che si differenzia dall'annotazione perché in alto a sinistra del box compare il suo nome, può essere collegata tramite una freccia che si aggancia al box di destinazione.

10.4.4 Aggiunta di un file multimediale dal Vault

se all'interno del Vault è presente un file multimediale è possibile inserirlo in un box all'interno della lavagna (pdf, link esterni a siti, immagini, ecc): da notare che se viene inserito un link ad un sito il box che ne deriva è navigabile all'interno di Obsidian e questo permette una libertà di creazione davvero ampia.

10.4.5 Aggiunta di collegamenti:



Fig. 10.4.5.1: aggiunta di collegamenti

E' possibile collegare tra loro i vari elementi creati attraverso delle frecce che possono essere modificate nel colore per meglio evidenziarle e aggiungendo una etichetta. La freccia che collega due box si crea nel seguente modo: si posiziona il cursore nel bordo del box di partenza fino ad evidenziare il pallino viola; si tiene premuto il tasto sinistro del mouse e si trascina fino al box destinazione rilasciando nella posizione desiderata. La situazione è mostrata in figura 10.4.5.1.

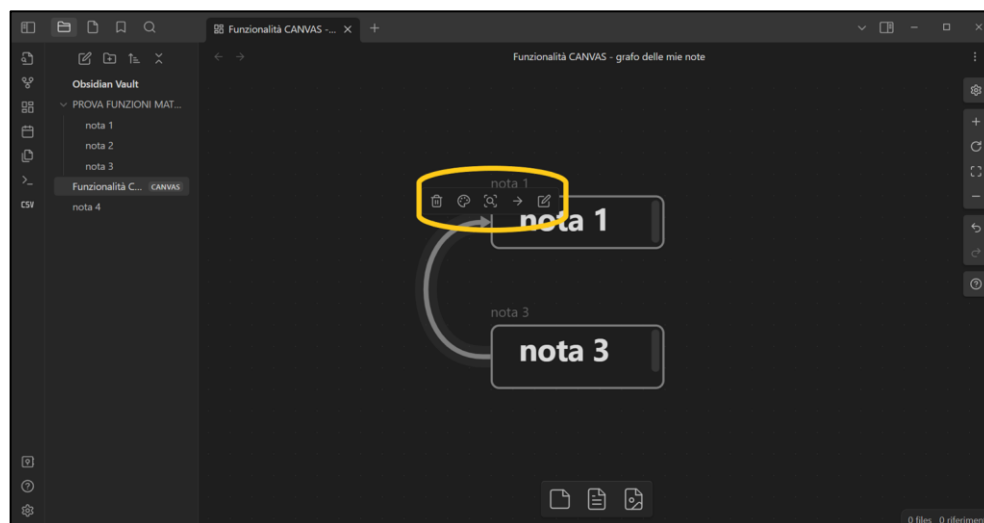


Fig. 10.4.5.2: box di funzionalità contestuale

Una volta creata la freccia questa è modificabile: cliccando sopra con il tasto sinistro del mouse si evidenzia l'elemento e appaiono le icone per le modifiche. In figura 10.4.5.2 si mostra ciò che appare a schermo. E' possibile cancellare l'elemento, impostare il colore, fare uno zoom sull'elemento, cambiare la direzione della freccia, rimuovere l'etichetta, modificare l'etichetta.

10.4.6 Funzionalità di Zoom e scorrimento

Per muoverci all'interno della nostra lavagna ci sono una serie di funzionalità per lo spostamento e lo zoom molto utili:

- per spostarsi verso destra e verso sinistra bisogna premere il tasto SHIFT o MAIUSC da tastiera tenerlo premuto e scorrere con la rotella del mouse;
- tenendo premuta la rotella del mouse si esegue uno spostamento panoramico;
- tenendo premuto il tasto CTRL o la barra spaziatrice e scorrendo la rotella del mouse si esegue uno zoom dinamico.

E' possibile ritornare ad una visualizzazione che si adatta alla finestra tramite il pulsante relativo che si trova tra le icone in alto a destra e che sono segnate in figura nel riquadro azzurro. Per fare uno zoom sulla nota di interesse basta cliccare sul box, il box diventa colorato in viola nel suo bordo e compaiono le icone in alto, e premere la combinazione di tasti SHIFT + 2 oppure cliccare sull'iconcina relativa del box.

10.4.7 Creazione di un gruppo

Altra funzionalità molto interessante è quella che permette la creazione di un gruppo. Per farlo si può creare un riquadro attorno alle note di interesse, comparirà un box con delle icone in alto e tra queste abbiamo l'icona di creazione del gruppo. Una volta creato si potrà nominarlo cliccando sopra con il tasto destro sulla posizione relativa. Si può creare il gruppo in modo alternativo: dopo aver selezionato le note di interesse facendo click con il tasto destro del mouse, nel menù a tendina che si genera, selezioniamo crea gruppo.

10.4.8 Esportazione della lavagna con un'immagine

Premendo il tasto con i tre pallini verticali in alto a destra si apre un menù a tendina. Selezioniamo esporta come immagine. Si apre una finestra di interazione che ci consente di esportare l'immagine di tutta la lavagna o solo dell'area visibile. Premendo il pulsante salva possiamo salvare il file in formato .png dove vogliamo all'interno del nostro PC. I file Canvas vengono archiviati localmente utilizzando un formato JSON che è stato progettato per funzionare con altri strumenti. App, script e plug-in possono facilmente migliorare Canvas aggiungendo o modificando schede e connessioni.

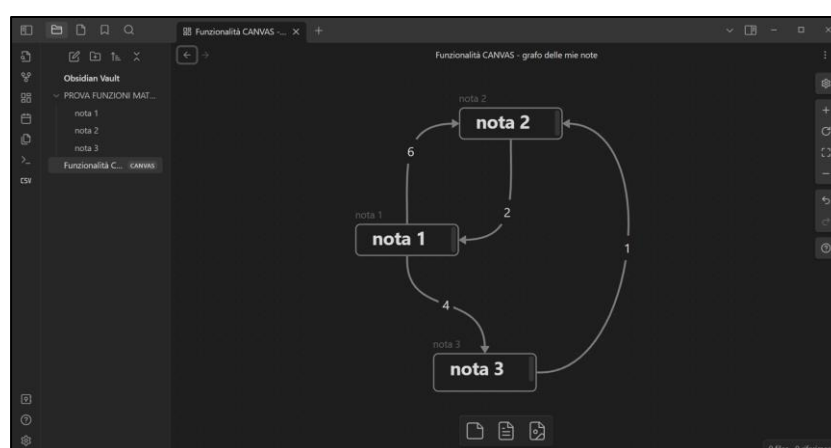


Fig. 10.4.8.1: esempio di note collegate e relativi “pesi”

10.4.9 Collegamento con il plugin “Adjacency Matrix Exporter”:

Canvas ci consente di effettuare, tramite una lavagna infinita, la creazione di un grafo e di utilizzare lo strumento di esportazione della matrice di adiacenza. E' sufficiente infatti trasformare i box delle annotazioni in note vere e proprie salvate dentro il Vault. Per far questo si fa click sul box per evidenziarlo, come mostrato in figura 10.4.9.1, poi click destro per aprire il menù a tendina da qui si

seleziona l'opzione converti in file. A questo punto l'annotazione diventa un file Markdown all'interno del Vault nell'esempio, mostrato in figura 10.4.9.2 e 10.4.9.3, ho creato la nota 4 e ho inserito 3 collegamenti alla nota 1 direttamente dal box. Attivando il plugin "Ajacency Matrix Exporter" la matrice di adiacenza esportata tiene conto della nota 4. E' quindi possibile operare tramite la funzionalità Canvas, convertire in grafo le annotazioni di interesse ed esportare la matrice CSV secondo le modalità spiegate in precedenza.

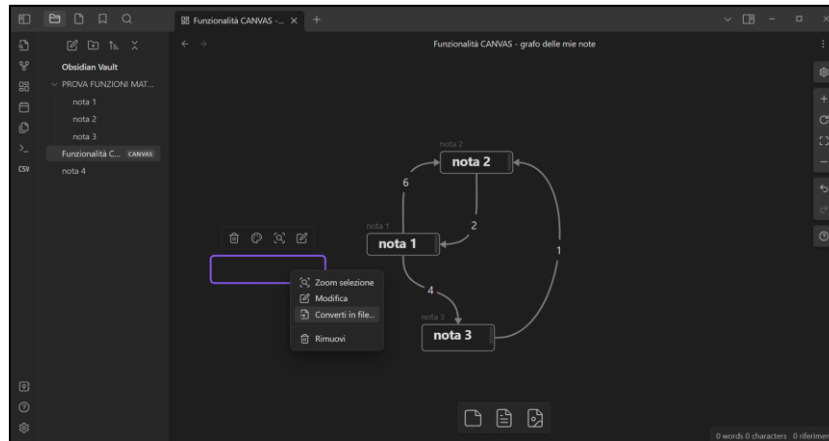


Fig. 10.4.9.1: azione di creazione della nota 4

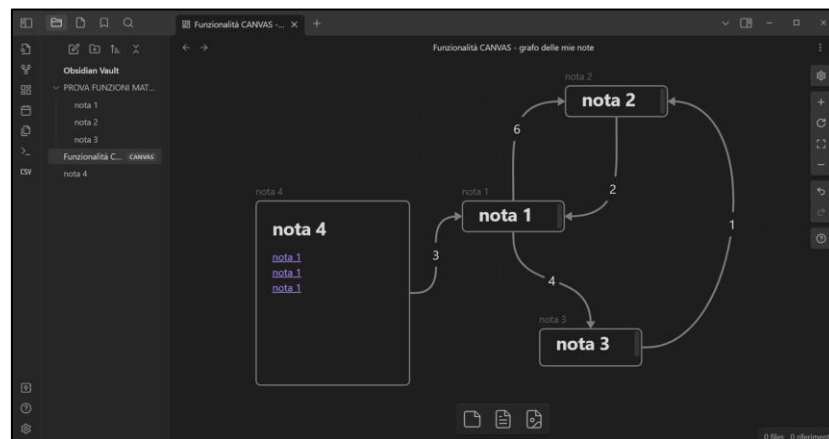


Fig. 10.4.9.2: si nomina la nota 4

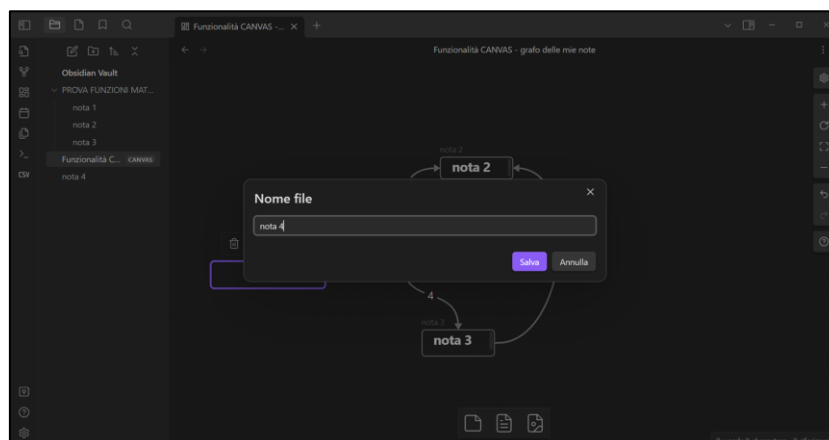


Fig. 10.4.9.3: la nota 4 è collegata al grafo della rete di esempio

11 Bibliografia:

Daniele Grazzini, *Sviluppo di un sistema di valutazione dell'efficienza nei processi automotive tramite la Design Structure Matrix: il caso del brevetto Seat Bridge*, tesi di laurea triennale Uninettuno Roma, 2024.

Fabio Manciocchi, *Design Structure Matrix: analisi dei modelli e metodi di struttura per sistemi di produzione*, tesi di laurea triennale Uninettuno Roma, 2024

Steven D. Eppinger, Tyson R. Browining, *Design Structure Methods and Application*, The MIT Press Cambridge, 2012.

Tyson R. Browining, *Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions*, IEEE Transactions on Engineering Management, Vol. 48, n° 3, 2001, 292-306.

Tyson R. Browining, Steven D. Eppinger, *Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development*, IEEE Transactions on Engineering Management, Vol. 49, n° 4, 2002, 428-442.

Tyson R. Browining, *Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities*, IEEE Transactions on Engineering Management, Vol. 63, n° 1, 2016, 27-52.

Maheswari J. U., *Scheduling Techniques in Projects*, IIT Delhi, 2021.

Stephen Denker, *Mastering Complexity*, CRC Press, 2015.

Amro M. Farid, Nam P. Suh, *Axiomatic Design in Large Systems*, Springer, 2016.

Qi Dong, *Predicting and Managing System Interactions at early phase of the product development process*, PHD Thesis, MIT, 2002.

Lindemann U., Maurer M., Brown T., *Structural Complexity Management*, Springer, 2009.

Wawge Anshul Rameshwar, Ashish P. Waghmare, *Analytical Tools Used For Finding The Delay In Bridge Construction Projects And Suggestions For Improvements-A Review*, International Journal of Progressive Research in Engineering Management and Science, 2023.

Yasin Affan, et al. , *Gamifying requirements: An empirical analysis of game-based technique for novices*, Journal of Software: Evolution and Process, 2017.

Fathullah, Muhammad Afif, Anusuyah Subbarao, Saravanan Muthaiyah, *Methodological Investigation: Traditional and Systematic Reviews as Preliminary Findings for Delphi Technique*, International Journal of Qualitative Methods 22, 2023.

Masato T., Yutaka N., Kikuo F., *Dendrogram-directed design space exploration for DSM-based product modularization*, JSME, Vol. 16, n° 5, 2022.

Batallas D. Andres, Yassine A. Ali, *Information Leaders I Product Development Organizational Networks: Social Network Analysis of the Design Structure Matrix*, IEEE Transactions on Engineering Management, Vol. 53, n° 4, 2006, 570-582.

Keller R., Eckert M. Claudia, Clarkson P. John, *Matrices or node-link diagrams: which visual representation is better for visualizing connectivity models ?*, Palgrave Macmillan, 2006, 62-76.

Umunnakwe A., Huang H., Oikonomou K., Davis K. R., *Quantitative analysis of power systems resilience: Standardization, categorizations and challenges*, Renewable and Sustainable Energy Reviews, Vol. 149, 2021.

Yassine A., Falkenburg D., Chelst K., *Engineering design management: An information structure approach*, International Journal of Production Research, Vol. 37, n° 13, 1999, 2957-2975.

Hamraz B., Caldwell N. H. M., Clarkson P. J., *A Holistic Categorization Framework for Literature on Engineering Change Management*, Systems Engineering, Vol. 16, n° 4, 2013, 473-506.

Hillier F. S., Hillier M. S., Schmedders K., Stephens M., *Introduction to Management Science*, McGraw-Hill, 2014.

TESTI UTILI PER L'APPRENDIMENTO DELLA PROGRAMMAZIONE TYPESCRIPT:

Freeman Adam, *Essential Typescript 4*, Apress, 2021.

Zametti Frank, *Modern Full-Stack Development*, Apress, 2020.

Goldberg Josh, *Learning Typescript*, O'Reilly, 2022

Ivanov M., Bespoyasov A., *Fullstack React with Typescript*, 2021.

Biswas N., *The Complete Typescript Guide with Projects*, Leanpub, 2022.

Uzayr S. B., *Typescript for Beginners*, CRC Press, 2022.

Ringraziamenti

A coronamento del lavoro svolto sento il desiderio di ringraziare coloro che hanno reso possibile il raggiungimento di questo obiettivo. La lista, che non rende giustizia ai numerosi contatti con i quali ho avuto scambi preziosi, sarebbe molto lunga ma mi limito a citare i più importanti senza però sminuire coloro che hanno avuto un ruolo anche piccolo ma determinante.

Ringrazio il titolare del corso di Impianti Industriali **Prof. Elpidio Romano** e il mio relatore della tesi di laurea triennale il **Prof. Andrea Falegnami** per avermi dato la possibilità di cimentarmi in un percorso stimolante e appassionante di ricerca sperimentale e per avermi ritenuto all'altezza di portarlo a compimento;

ringrazio l'Università Uninettuno nella persona del Rettore **Prof.ssa Maria Amata Garito** per aver creato un ambiente di studio estremamente positivo fondato su un livello alto di contenuti e su un team di persone capaci e competenti;

ringrazio i professionisti che hanno partecipato alla raccolta dati della mia ricerca sperimentale e che ritornano protagonisti di questo breve lavoro di compendio: hanno costituito il gruppo ESM che ho avuto l'onore e il piacere di condurre attraverso il percorso di ricerca effettuato per la loro professionalità, entusiasmo, estrema disponibilità e collaborazione dimostrata;

ringrazio il **Prof. Tyson R. Browning** (Christian Texas University) con il quale sono entrato in contatto e che mi ha fornito spunti e materiali preziosi per approfondire i temi di ricerca sul DSM e sugli strumenti utilizzati;

ringrazio il Sig. **Ross Keenan** autore del plugin "Adjacency Matrix Maker" per aver sostenuto ed apprezzato il mio lavoro di programmazione e aver condiviso con me il suo;

ringrazio quanti hanno contribuito a titolo amichevole a rendere più ricca e appassionante questa ricerca ed in particolare il **Prof. Gabriele Menna**, il **Sig. Taras Kushnir**, il **Dr. Fabio Manciocchi**, il professionista **Dr. Bashkim Zhegrova**, gli ingegneri del software **Dr. Jhoannes Theiner** e il **Dr. Liam Cain**;

ringrazio la mia famiglia e i miei familiari tutti per avermi sostenuto, incoraggiato, assistito e sopportato;

ringrazio il mio titolare **Sig. Americo Innocenzi** e i figli **Leonardo e Tommaso Innocenzi** che hanno creduto nel mio impegno portato avanti con fatica ma con determinazione;

ringrazio il **Sig. Stefano Santarelli**, amico carissimo che ha sostenuto questo progetto formativo e ha creduto nelle possibilità di miglioramento che avrebbe offerto;

ringrazio infine il **Dr. Sandro Zilli** professionista consulente e formatore, fautore del principio del Lifelong Learning la cui influenza positiva mi accompagna dal conseguimento del diploma ITIS in Elettronica ed Elettrotecnica.