



UNIVERSITA' DEGLI STUDI DI PADOVA

FACOLTA' DI INGEGNERIA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

CORSO DI LAUREA IN INGEGNERIA GESTIONALE

Tesi di laurea di primo livello

“Modello di screening neonatale per malattie metaboliche: trascrizione in linguaggio Arena”

RELATORE:

CHIAR.MO PROF. Giorgio Romanin Jacur

LAUREANDO: Michele Di Pieri

MATRICOLA N. 592934

ANNO ACCADEMICO 2010 – 2011

INDICE

Introduzione: La simulazione.....7

Capitolo 1: Il software Arena.....8

1.1 Modulo Create.....8

1.2 Modulo Dispose.....9

1.3 Modulo Process.....9

1.4 Modulo Decide.....10

1.5 Modulo Assign.....10

1.6 Modulo Record.....10

Capitolo 2: Il modello: panoramica generale.....11

2.1 Analisi dei dati storici.....13

2.1.1 Dati relativi alle nascite.....13

2.1.2 Dati relativi ai risultati positivi.....14

2.1.3 Dati relativi alle distribuzioni temporali.....14

Capitolo 3: Il modello di screening neonatale.....15

3.1 Settore 1: Generazione, prelievo e spedizione.....	17
3.1.1 Generazione nascite.....	17
3.1.2 Prelievo.....	18
3.1.3 Spedizione.....	20
3.2 Settore 2: Centro di primo livello, Padova.....	22
3.2.1 Primo test, Padova.....	22
3.2.2 Secondo test, Padova.....	24
3.3 Settore 3: Centro di primo livello, Verona.....	25
3.4 Settore 4: Risultati dei test.....	25
3.5 Settore 5: Risultati negativi e secondo prelievo....	28
3.6 Settore 6: Test di secondo livello e raccolta dati...	29
3.6.1 Centro di secondo livello.....	30
3.6.2 Raccolta dati.....	31
3.7 Settore 7: Uscita.....	32

Conclusioni.....33

Bibliografia.....34

INTRODUZIONE: La simulazione

Il lavoro consiste nel creare un modello di simulazione per lo screening neonatale nel linguaggio Arena[®], partendo da una base realizzata in linguaggio Micro Saint[®], a causa dell'incompatibilità di quest'ultimo con i più recenti sistemi operativi. Entrambi questi linguaggi di programmazione si basano sulla logica della simulazione, ovvero la creazione su appositi software di un modello della realtà che consenta di valutare e prevedere lo svolgersi di una serie di eventi. La simulazione è, quindi, la trasposizione in termini logico – matematico – procedurali di questo “modello concettuale” della realtà, rappresentante l'insieme dei processi che hanno luogo nel sistema studiato e il cui insieme permette di valutare le logiche di funzionamento del sistema stesso. Perciò si è diffuso l'utilizzo della simulazione nell'analisi di vari sistemi: dinamici (semplici o complessi), naturali (come per esempio il sistema climatico), artificiali (sistemi meccanici) o socio-economici.

Il vantaggio della simulazione consiste nel poter prevedere diversi scenari, con il cambiamento di poche variabili, senza dover creare fisicamente ognuno di essi; inoltre viene valutata anche la dipendenza dal tempo delle variabili. Tra gli altri benefici derivanti dal suo utilizzo, si possono contemplare la riduzione dei rischi, dei costi e dei tempi; la velocità di adattamento a variazioni del modello; una maggior comprensione finale del fenomeno.

La simulazione può essere articolata in 8 fasi:

1. identificazione del problema da esaminare e degli obiettivi: vengono definite le relazioni sulle quali concentrarsi e le informazioni da fornire;
2. analisi del sistema: si decide il livello di dettaglio a cui scendere nel modello, in quanto non è possibile realizzare un modello perfetto che comprenda ogni più piccolo particolare;
3. raccolta e analisi dei dati: vengono raccolti e analizzati statisticamente i dati raccolti;
4. costruzione del modello: sulla base dei dati raccolti, si identificano le entità del modello e le relazioni funzionali tra i vari processi che lo compongono;
5. stesura del programma al calcolatore: il modello viene codificato su un opportuno programma di simulazione, e viene verificato il suo funzionamento;
6. validazione del modello: viene controllato se il modello è aderente al modello reale;
7. conduzione degli esperimenti: si valuta in che modo il modello risponde ai vari scenari ipotizzati;
8. analisi dei risultati.

In questo lavoro ci si concentrerà soprattutto sulle fasi 4 e 5, ovvero la costruzione del modello su software simulativo Arena.

CAPITOLO 1: Il software Arena

Il software Arena prevede la possibilità di creare una larga varietà di modelli con ampia flessibilità riguardo al livello di dettaglio e complessità desiderato. La modellazione viene realizzata attraverso un diagramma di flusso a blocchi, collegati tra loro in base alle relazioni esistenti; ogni blocco è costitutivo di un modulo. L'elenco dei moduli si può trovare nella project bar del programma. Limitatamente a questo studio, basteranno i moduli individuati dall'intestazione "Basic Process".

Gli oggetti della simulazione sono le entità; un'entità può essere un oggetto, una persona, un documento; è ciò che fluisce attraverso il modello. Nei modelli di simulazione, ogni entità è definita univocamente; in un modello, possono anche venire create entità di tipo diverso. In Arena, le entità vengono create nel modulo "Create", seguono il diagramma lungo gli altri moduli, fino a uscire nel modulo "Dispose". Alle entità possono venire assegnati degli attributi: un attributo è una

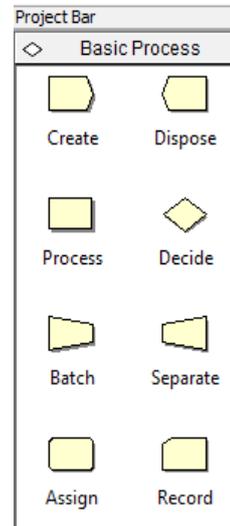


Figura 1: Basic Process

caratteristica di tutte le entità create, ma con specifici valori che possono variare da un'entità a un'altra. Ogni attributo ha un suo nome univoco, un valore iniziale, e può essere o un numero reale o una stringa. Il suo valore può variare nel modulo "Assign"; inoltre agli attributi ci si può riferire anche in altri moduli, come per esempio nel "Decide", e possono anche venire usati in ogni espressione all'interno del modello.

Il valore degli attributi è quindi unico per ogni entità, al contrario delle variabili il cui valore è globale nel modello. Come gli attributi, le variabili sono caratterizzate da un nome univoco nel modello, da valore iniziale e tipologia di dato (reale o stringa); il loro valore può variare nel modulo "Assign" e ci si può riferire ad esse in altri moduli o espressioni.

1.1 Modulo Create

Il modulo "Create" serve a generare le entità. Costituisce l'inizio del modello; una volta generate, le entità lasciano il modulo per proseguire nel modello. In questo modulo vengono definiti il tipo di entità, il tempo tra gli arrivi, il numero di entità per arrivo. Per quanto riguarda il tempo tra gli arrivi, può essere di tipo

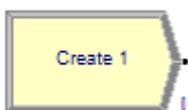


Figura 2: Modulo Create

Random (tempo che segue una distribuzione esponenziale con media definita dall'utente), Constant (tempo costante) o Expression (il tempo segue una distribuzione tra quelle presenti in una lista).

Il campo "Value" da' appunto il valore della media nella distribuzione Random, o della costante di tempo nella distribuzione Constant; nel campo "Units" si determina l'unità di tempo (giorni, ore, minuti...). Il numero di entità per arrivo e il numero massimo di entità che possono venir processate viene specificato nei due campi seguenti ("Entities per Arrival" e "Max Arrivals").

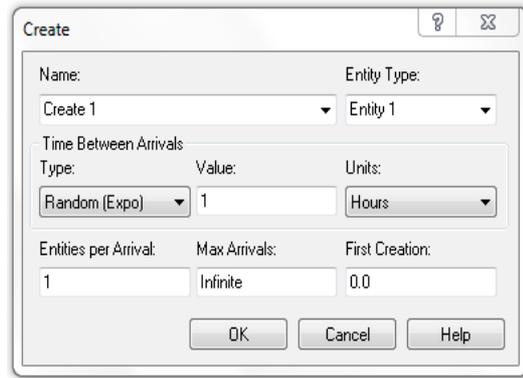


Figura 3: Modulo Create - Particolare

1.2 Modulo Dispose



Il modulo "Dispose" è il punto finale del modello. Una volta giunte a questo modulo, le entità escono dal modello e cessano di esistere.

Figura 4: Modulo Dispose

1.3 Modulo Process

Il modulo "Process" serve a simulare i vari processi che avvengono nel modello. I processi possono richiedere l'utilizzo di risorse, a cui sono associate code, dipendenti della limitata capacità della risorsa stessa.



Ci sono quattro tipi di processo: Delay, Seize Delay, Seize Delay Release, Delay Release. Il primo tipo è un processo in cui le entità subiscono solo ritardo, senza occupare risorse; il secondo è un processo di riempimento e attesa: la capacità della risorsa viene impiegata, e si presuppone ci sarà più avanti un processo di tipo Delay Release, in cui la risorsa verrà liberata; infine, il Seize Delay Release comporta occupazione della risorsa, suo utilizzo e successivo rilascio. Viene poi indicato il tipo di distribuzione temporale del processo: normale, triangolare, costante, uniforme oppure di tipo "Expression", indicata cioè da un'espressione algebrica. A seconda del tipo di distribuzione, si

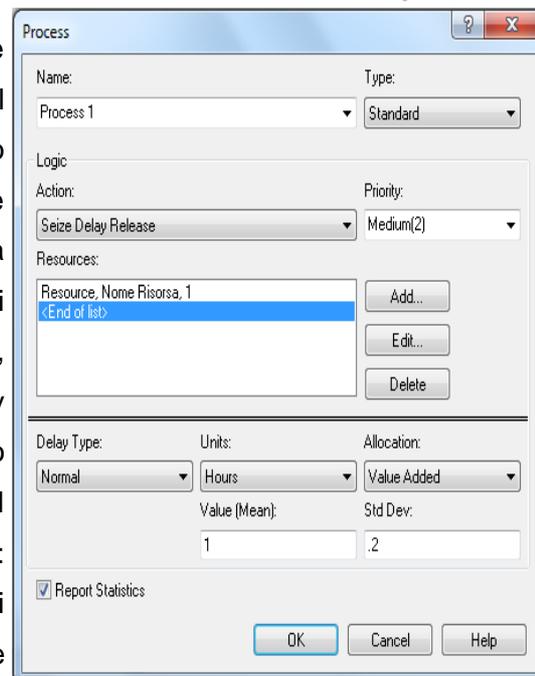
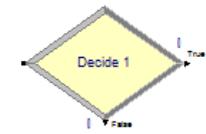


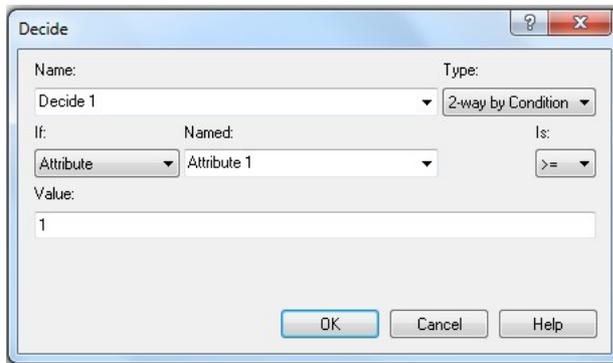
Figura 5: Modulo Process e particolare

indicano i vari parametri di tempo (media, deviazione standard, valori massimo e minimo).

1.4 Modulo Decide



Il modulo “Decide” serve nei casi in cui si hanno decisioni di tipo “what-if”, o casi in cui uno scenario avvenga solo in presenza di una certa variabile o attributo. Ci possono essere 4 tipi di decisione: 2-way by chance, 2-way by condition, N-way by chance, N-way by condition, a seconda se le possibilità sono due



(2-way, casi ad esempio di vero/falso) o più di due (N-way). Se si è nella situazione “by chance”, le decisioni sono influenzate da una percentuale definita; se si è nella situazione “by condition”, è un particolare valore di una variabile, di un attributo, di un'espressione, o il tipo di entità, a determinare la decisione.

Figura 6: Modulo Decide e particolare

1.5 Modulo Assign

Il modulo “Assign” serve ad assegnare nuovi valori alle variabili, al tipo di entità, o agli attributi di un'entità. In un modulo si possono fare anche assegnazioni multiple, di più parametri diversi.

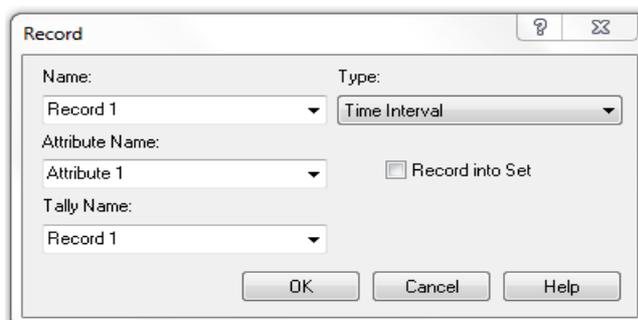


Figura 7: Modulo Assign

1.6 Modulo Record



Il modulo “Record” è utilizzato per la raccolta delle statistiche del modello. Nel modulo sono possibili diversi tipi di statistiche; in questo lavoro verranno utilizzate solo i tipi Count e Time Interval. Il primo aumenta la statistica in questione di un valore determinato dall'utente; suo tipico uso sono i contatori. Il secondo calcola e



immagazzina la differenza tra il valore di uno specifico attributo e il tempo di simulazione corrente; come risultato finale, da' la media, il valore massimo e il valore minimo tra tutte le entità processate da questo modulo. Questi valori vengono esplicitati in un “tally”, a cui normalmente si da' per comodità lo

Figura 8: Modulo record e particolare
stesso nome del modulo.

CAPITOLO 2: Il modello: panoramica generale

Le malattie metaboliche ereditarie sono rare, ma possono produrre gravi danni se non vengono curate in tempo. Si è dunque pensato di realizzare un programma di screening neonatale nella regione Veneto, inclusivo di un primo test alla nascita e di un secondo test più accurato per i pazienti che risultino positivi. Lo screening consiste nel fare un test a tutti i soggetti di una popolazione con lo scopo di identificare una malattia prima che ne compaiano i sintomi. In caso di risultato positivo, si hanno in seguito una diagnosi più accurata e le necessarie terapie.

Il progetto comprende unicamente la regione Veneto; in essa sono presenti due centri operativi di primo livello, situati a Verona e Padova, a cui fanno riferimento anche le altre province (Vicenza si riferisce a Verona; Venezia, Belluno, Treviso e Rovigo a Padova). Inoltre, sempre a Padova, è situato un centro di secondo livello, dove vengono inviati i neonati a cui viene diagnosticata una delle patologie ricercate. Obiettivo del progetto è dimensionare ottimamente la capacità dei due centri di primo livello e di quello di secondo, cioè nello specifico il numero di controlli effettuati giornalmente dai primi e numero di test eseguiti a settimana per il secondo. La capacità andrà decisa in modo da non imporre ritardi pericolosi nell'analisi dei prelievi e nella cura dei pazienti risultati positivi, ma senza richiedere eccessive spese con la creazione di centri sovradimensionati.

Le azioni relative allo screening neonatale sono le seguenti: viene fatto un prelievo ad ogni neonato entro 2-3 giorni dalla nascita; questo prelievo è poi inviato velocemente al centro di primo livello (Padova o Verona), dove viene controllato non appena è possibile; se la prima misurazione da' un risultato dubbio (borderline), si procede a un nuovo test dello stesso campione; se il risultato non è certo, si procede immediatamente a un nuovo prelievo, ma se è relativo a una nascita prematura il nuovo prelievo è fatto dopo 15 giorni; se il risultato è chiaro e risulta essere positivo, il neonato è trasferito al centro di secondo livello a Padova, dove viene effettuato un nuovo test per ottenere una diagnosi più accurata e si programmano le terapie. Questo secondo test è fatto non appena possibile.

Il modello viene sviluppato nell'arco temporale di un anno, con warm-up iniziale di un altro anno.

Per ogni patologia ricercata, verrà dato il nome ufficiale, l'abbreviazione comune e la priorità. La priorità più alta consiste nel simbolo +++, e va a indicare che per tali patologie saranno da iniziare le terapie entro la prima settimana di vita; a seguire si ha priorità ++, con patologie che richiedono la cura entro le prime due settimane di vita, e priorità +, per patologie che non richiedono terapie entro le prime due settimane (verrà indicato entro le prime tre settimane per non avere ritardi eccessivi).

Le diverse patologie sono presentate nella tabella seguente.

Nome scientifico e abbreviazione	Priorità
Phenylketonuria (PKU)	++
Tyrosinemia, Type I (TYR I)	+++
Citrullinemia (CIT)	+++
Argininosuccinic Aciduria (ASA)	+++
Arginase Deficiency (ARG)	+++
Maple Syrup Urine Disease (MSUD)	+++
Propionic Acidemia (PA)	+++
Methylmalonyl-CoA Mutase (MMA)	+++
Methylmalonic acidemia with homocystinuria type C/D (Cbl C/D)	+++
Isovaleric acidemia (IVA)	+++
Glutaric aciduria I (GA I)	++
Very long chain acyl-CoA dehydrogenase deficiency (VLCAD)	+++
Long chain 3-hydroxyacyl-CoA dehydrogenase deficiency (LCHAD)	+++
Medium chain acyl-CoA dehydrogenase deficiency (MCAD)	++
Short chain acyl-CoA dehydrogenase deficiency (SCAD)	+++
Glutaric aciduria 2 (GA II)	++
Carnitine palmitoyl transferase 1 deficiency (CPT I)	++
Carnitine palmitoyl transferase 2 deficiency (CPT II)	+++
Carnitine-acylcarnitine translocase deficiency (CACT)	+++
Carnitine uptake deficiency (CUD)	++
Glycogen storage disease type II (Pompe disease) (GSD II)	++
Fabry Disease (FD)	+
Mucopolysaccharidosis I,II,VI (MPS I, II, VI)	+

Tabella 1: malattie che richiedono lo screening

2.1 Analisi dei dati storici

In questo paragrafo verranno analizzati i dati storici richiesti per la realizzazione del modello. Tra questi, i dati relativi alle nascite, alla percentuale di risultati positivi, alle stime delle distribuzioni temporali.

2.1.1 Dati relativi alle nascite

I dati riguardanti le nascite sono stati ottenuti dai certificati di nascita locali dell'anno 2009. Uno studio accurato ha mostrato come il numero di nascite dipenda dal mese dell'anno e dal giorno della settimana. Dunque per la stima delle nuove nascite si ricorre a un modello di regressione lineare dato dalla formula $\mu_{ij} = \alpha + \beta_i + \delta_j$, con $i = 1, \dots, 12$ a rappresentare i mesi e $j = 1, \dots, 7$ a rappresentare i giorni della settimana. I coefficienti α , β , δ sono parametri conosciuti grazie ai dati storici.

PARAMETRO	VALORE	PARAMETRO	VALORE
α	105,50	β_{10}	2,45
β_1	0,00	β_{11}	-6,13
β_2	-1,57	β_{12}	-3,59
β_3	-7,78	δ_1	37,70
β_4	-7,39	δ_2	30,97
β_5	-3,90	δ_3	37,51
β_6	-2,32	δ_4	26,24
β_7	7,15	δ_5	34,99
β_8	2,86	δ_6	8,22
β_9	10,42	δ_7	0,00

Tabella 2: valori dei coefficienti

Inoltre, la proporzione delle nascite rispetto ai due centri è del 39% per Verona e 61% per Padova.

2.1.2 Dati relativi ai risultati positivi

Per quanto concerne la percentuale di prelievi risultati positivi per le varie patologie, sono stati ottenuti in parte grazie agli screening realizzati in altre regioni o Stati, in parte grazie a studi clinici (Da Frè 2011). I valori ottenuti sono diversi, e per motivi di sicurezza relativi ai neonati verrà considerato il caso peggiore, cioè quello con percentuali massime. Alcune patologie sono state accorpate insieme in quanto il loro marcatore è lo stesso. Infine verranno accorpate per priorità, in quanto sarà così che verranno analizzate nel modello.

PATOLOGIA	% POSITIVITÀ	PATOLOGIA	% POSITIVITÀ
PA+MMA+Cbl C/D	0,0840%	MCAD	0,0283%
IVA	0,0213%	SCAD	0,0254%
GA I	0,0248%	GA II	0,0036%
PKU	0,0692%	CPT I	0,0016%
TYR I	0,0256%	CPT II+CACT	0,0192%
MSUD	0,0113%	CUD	0,0356%
ARG	0,0057%	GSD II	0,0390%
ASA+CIT	0,0122%	FD	0,3800%
VLCAD	0,0132%	MPS	0,0250%
LCHAD	0,0048%		

Tabella 3: % di neonati positivi in base alla patologia– dati storici

PRIORITÀ	% POSITIVITÀ
1	0,4050%
2	0,2021%
3	0,2227%

Tabella 4: % di neonati positivi in base alla priorità – dati storici

Inoltre, la percentuale di test ripetuti dovuti a risultati borderline è pari al 10%, mentre la percentuale di prelievi ripetuti per nascite premature è pari al 2%.

2.1.3 Dati relativi alle distribuzioni temporali

I dati relativi alla stima delle distribuzioni temporali saranno utilizzati nel modello solo nei moduli di tipo Process, in quanto è in questi moduli che si hanno le attese o le code. Il tempo passato tra la nascita e il prelievo è stimato con una distribuzione rettangolare (Uniform) tra 2 e 3 giorni; la spedizione del prelievo a uno dei due centri è a sua volta stimata con una distribuzione rettangolare tra 0,2 giorni e 1 giorno (tra 4,8 e 24 ore quindi); i centri di primo livello impiegano esattamente 1 giorno a ottenere i risultati, mentre quello di secondo livello impiega 0,625 giorni (15 ore). Infine, se c'è bisogno di un nuovo prelievo, c'è un'attesa di 15 giorni. A questi ultimi valori corrispondono quindi distribuzioni di tempo costanti.

CAPITOLO 3: Il modello di screening neonatale

In questo capitolo verrà presentata nella sua interezza il modello di screening neonatale, costruito sul software Arena. Per la sua creazione, sono stati utilizzati i moduli descritti in precedenza: Create, Dispose, Process, Decide, Assign, Record. Il modello verrà percorso da un unico tipo di entità, chiamato “Newborn”. A queste entità sono associati sette diversi attributi:

- coda_tot;
- prel;
- coda_1;
- centro;
- flag;
- prior;
- coda_2.

Il loro preciso significato verrà specificato in seguito, quando essi compariranno nel modello. Tutti e sette gli attributi hanno valore iniziale pari a 0.

Sono inoltre presenti tre risorse, corrispondenti ai due centri di primo livello e a quello di secondo. Per ognuno di essi va fissata la capacità. Poichè questo studio è concentrato sulla costruzione del modello e non sull'analisi dei risultati, verranno utilizzati i valori ottimi riscontrati dal modello originario in Micro Saint: capacità pari a 125 per il centro di primo livello a Padova, 85 per quello di Verona, 10 per il centro di secondo livello.

Infine, verranno utilizzate due variabili, chiamate “giorno” e “mese”, a indicare i coefficienti δ e β nella formula delle nascite. I loro valori sono riportati nelle seguenti figure.

Variable - Basic Process							
	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values
1 ▶	giorno	7	1	Real	System		7 rows
2	mese	12	1	Real	System		12 rows

Initial Values	
	1
1	37.7
2	31.0
3	37.5
4	26.2
5	35.0
6	8.2
7	0.0

Initial Values	
	1
1	0
2	-1.6
3	-7.8
4	-7.4
5	-3.9
6	-2.3
7	7.2
8	2.9
9	10.4
10	2.5
11	-6.1
12	-3.6

Figure 9, 10, 11: le variabili nel modello e loro valori iniziali

Qui di seguito è riportata, infine, la realizzazione del modello. A causa della sua grandezza, esso verrà analizzato a settori, e per ogni settore verranno presentati i moduli che lo compongono.

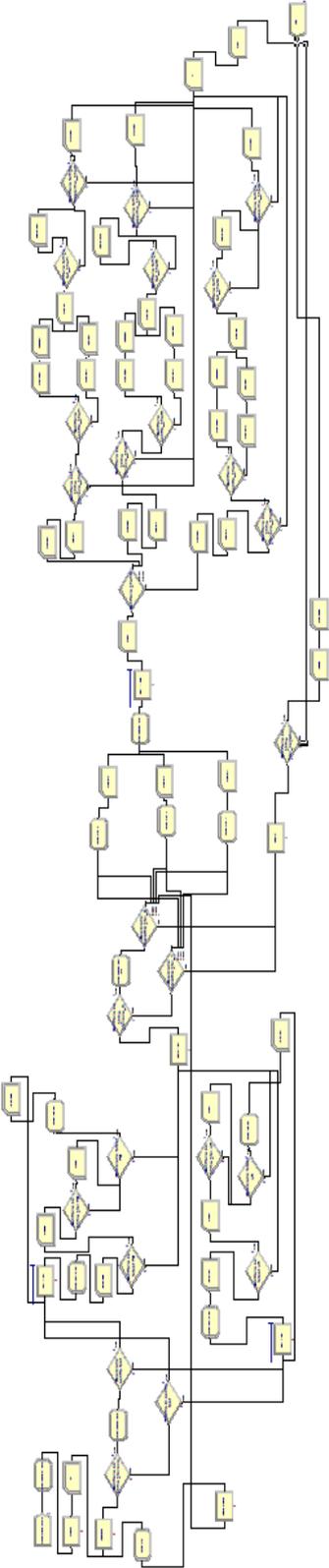


Figura 12: Il modello di screening neonatale

3.1 Settore 1: Generazione, prelievo e spedizione

Il primo settore comprende l'inizio del modello, dalla generazione casuale delle nascite alla divisione dei prelievi tra i vari centri. Esso comprende 9 moduli. I collegamenti incompleti rimandano a moduli di altri settori.

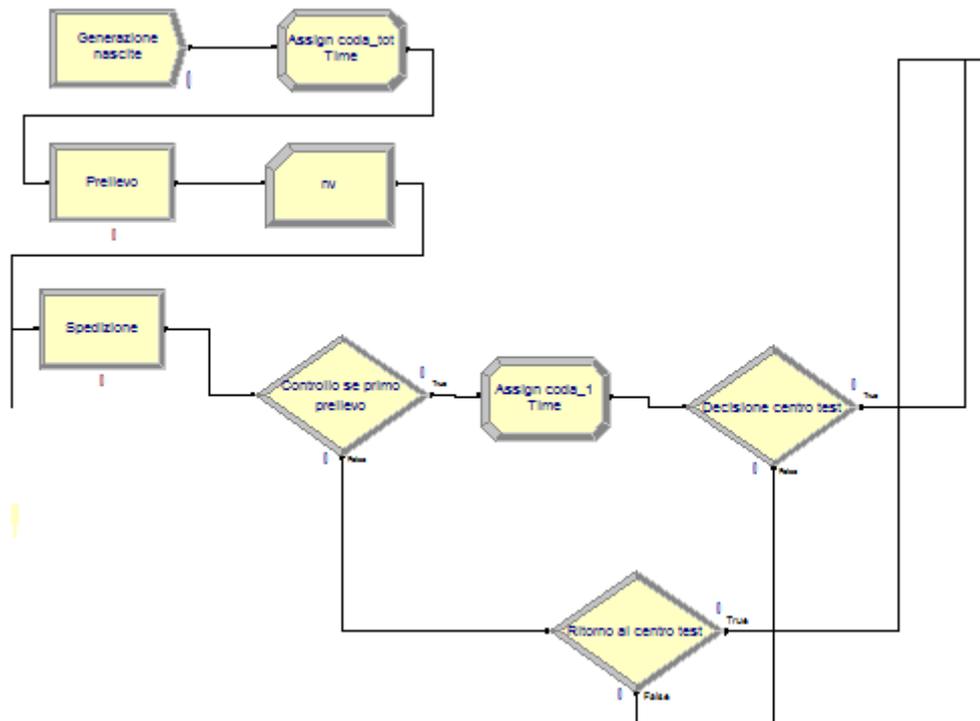


Figura 13: Settore 1

3.1.1 Generazione nascite

Il modulo "Generazione nascite" è di tipo "Create". Esso genera le entità (di tipo Newborn, come già detto) con distribuzione temporale di tipo esponenziale. Il valore della funzione è definito dall'espressione

$$1/(105.5+\text{giorno}(\text{CalDayOfWeek}(\text{TNOW}),1)+\text{mese}(\text{CalMonth}(\text{TNOW}),1)).$$

In essa sono utilizzate tre variabili predefinite di Arena: TNOW, CalDayOfWeek, CalMonth. TNOW indica il tempo corrente di simulazione; CalDayOfWeek da' come output un numero intero, compreso tra 1 e 7, che rappresenta il giorno della settimana corrispondente al tempo di simulazione; CalMonth è simile, con la differenza che il numero intero è compreso tra 1 e 12 e rappresenta il mese corrente. Dunque, l'espressione $\text{giorno}(\text{CalDayOfWeek}(\text{TNOW}),1)$ restituisce il valore nella tabella della variabile giorno presente nella riga CalDayOfWeek (colonna 1); allo stesso modo, l'espressione $\text{mese}(\text{CalMonth}(\text{TNOW}),1)$ restituisce il valore nella tabella della variabile mese presente nella riga CalMonth (colonna 1).

In una giornata, ci sono quindi μ arrivi, e ogni arrivo è costituito da una singola entità.

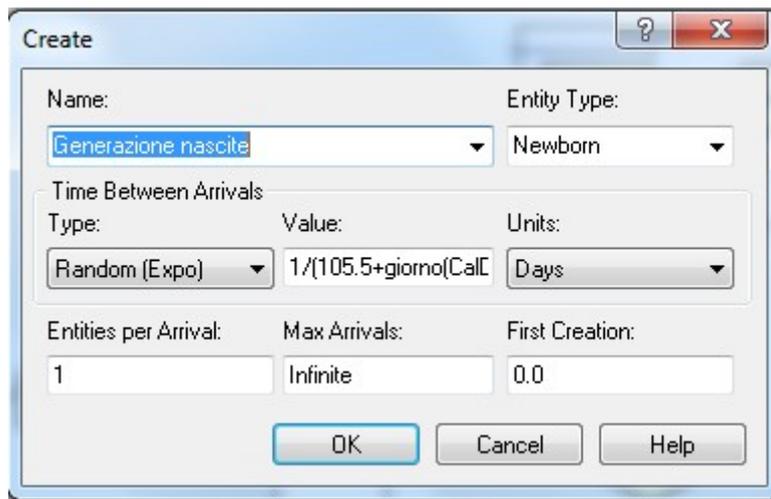


Figura 14: Generazione nascite

Il modulo successivo è il modulo “Assign coda_tot Time”, di tipo “Assign”. In esso, a ogni entità, viene associato il valore TNOW all'attributo coda_tot. Questo attributo servirà a valutare il tempo complessivo di percorrenza del modello, dalla nascita all'uscita.

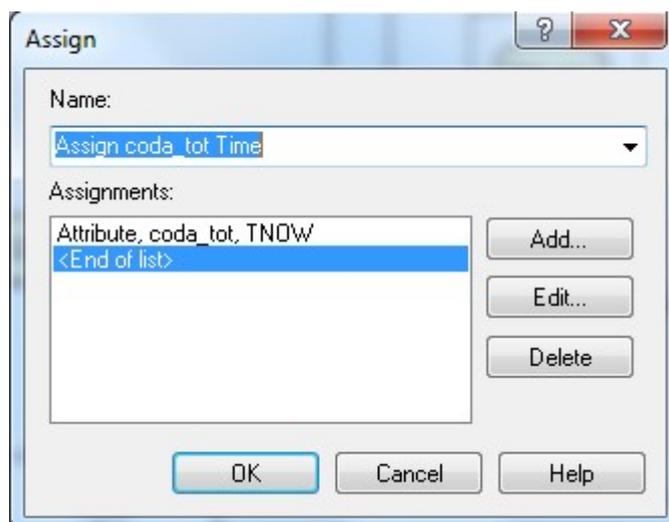


Figura 15: Assign coda_tot Time

3.1.2 Prelievo

Dopo la nascita, sui neonati viene eseguito il primo prelievo; questo avviene nel modulo “Prelievo”, di tipo “Process”. Esso è un processo di tipo “Delay”: infatti ad esso non corrisponde un utilizzo delle risorse evidenziate dal modello, ma altre di cui si suppone ci sia capacità sempre sufficiente (come possono essere infermieri, aghi, provette, ...). Come definito in precedenza, il tempo tra la nascita e il prelievo ha distribuzione rettangolare tra 2 e 3 giorni.

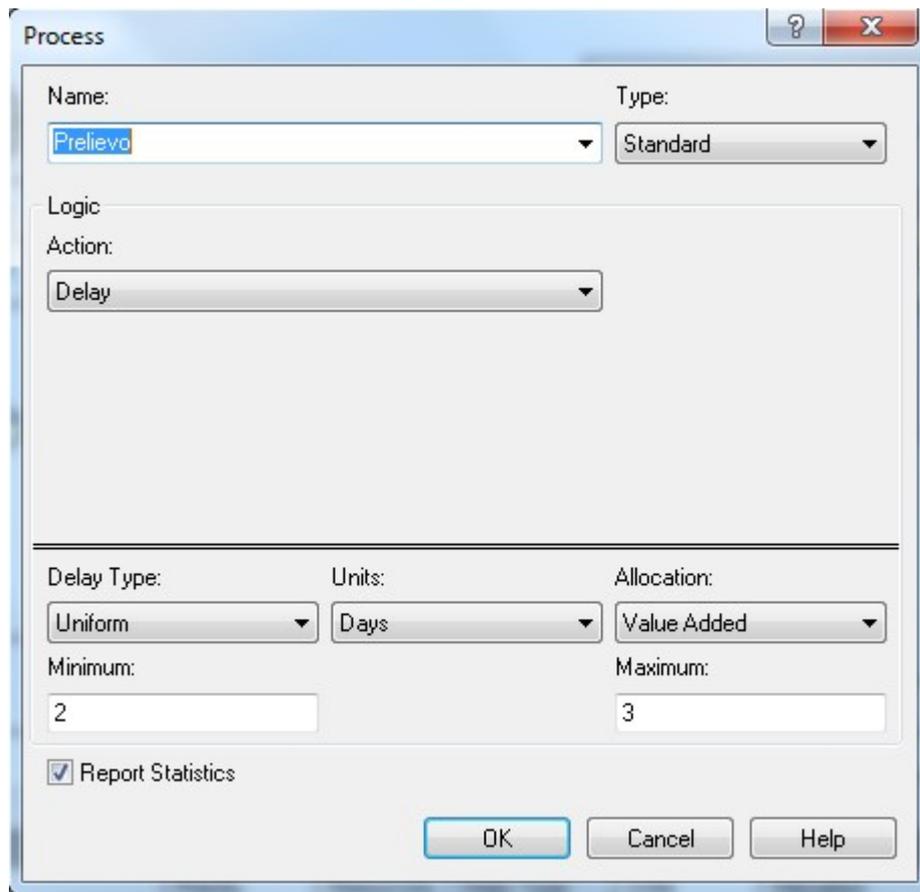


Figura 16: Prelievo

A seguire il prelievo, è presente il modulo “nv”, di tipo “Record”. Esso è di tipo contatore: suo compito è, infatti, tenere il conto delle nascite. In esso si può quindi vedere, alla fine della simulazione, il numero totale di entità processate. Il campo “Value” è quindi occupato dal numero 1: ogni entità che attraversa il modulo incrementa il valore di “nv” di 1.

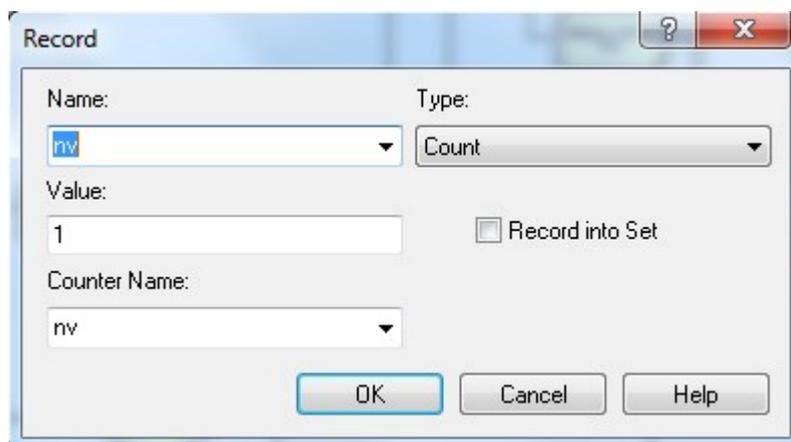


Figura 17: nv

Tutti i contatori hanno la stessa struttura di questo appena presentato; nel seguito quindi ci si limiterà a nominarli, senza presentarli in modo particolareggiato.

3.1.3 Spedizione

Dopo il prelievo, si ha la spedizione verso uno dei due centri di primo livello, a seconda della provincia in cui avviene la nascita. Il processo di spedizione è sempre di tipo "Delay", con distribuzione rettangolare, questa volta tra i valori 0.2 e 1.

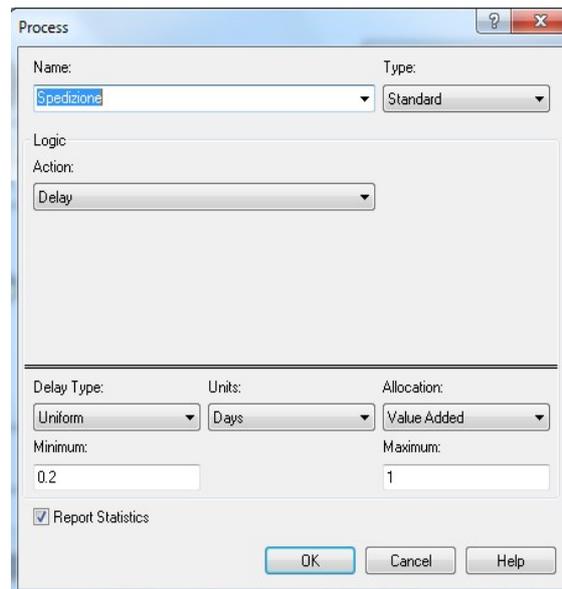


Figura 18: Spedizione

Alla spedizione giunge un'altra freccia, che verrà in seguito analizzata, proveniente dai prelievi rifatti per nascita prematura; le entità provenienti da questa parte di modello sono già state processate in uno dei due centri, e dunque esse dovranno tornare lì (per la definizione della divisione dei prelievi tra i centri basata sulle province). Perciò, dal modulo spedizione, si ha un primo modulo "Decide" che controlla se l'entità entrante ha già fatto o meno un secondo prelievo. Tale controllo viene fatto grazie all'attributo "prel": se "prel" vale 0, allora si tratta di primo prelievo, e il risultato è true (l'entità segue la freccia a destra); se "prel" è diverso da 0, allora si tratta di secondo prelievo e il risultato è false (l'entità segue la freccia in basso).

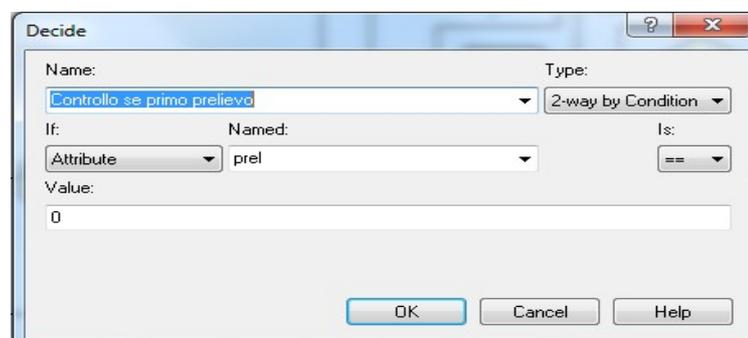


Figura 19: Controllo se primo prelievo

Come detto, seguendo la freccia in basso, l'entità deve tornare al centro di primo livello in cui era già stata processata in precedenza. L'attributo identificativo dei centri di primo livello è "centro". Se esso ha valore pari a 1, si tratta del centro a Padova; se pari a 2, quello di Verona. Dunque il modulo "Ritorno al centro test" si basa su una decisione 2-way by condition simile alla precedente; la differenza sta nell'attributo, che è "centro" al posto di "prel". Se il valore di "centro" è pari a 1, l'entità torna a Padova (valore true, freccia a destra); se è diverso da 1, l'entità torna a Verona (valore false, freccia in basso). Nei casi di decisione tra due alternative, si avrà sempre per il valore true la freccia destra, e per il valore false la freccia in basso; ciò non verrà più specificato nel prosieguo della spiegazione.

Ritornando al "Controllo se primo prelievo", nella parte true si passa a un nuovo modulo "Assign", che fissa il valore dell'attributo coda_1 con il tempo corrente di simulazione TNOW. Come il precedente attributo coda_tot, anche coda_1 serve a verificare i tempi di percorrenza, in questo caso a partire dall'arrivo al centro di primo livello.

Se si ha primo prelievo, la decisione sul centro a cui spedire il prelievo è effettuata tramite percentuale, come definita in precedenza; si ha dunque un modulo "Decide" di tipo 2-way by chance, con true corrispondente a Padova e false a Verona.

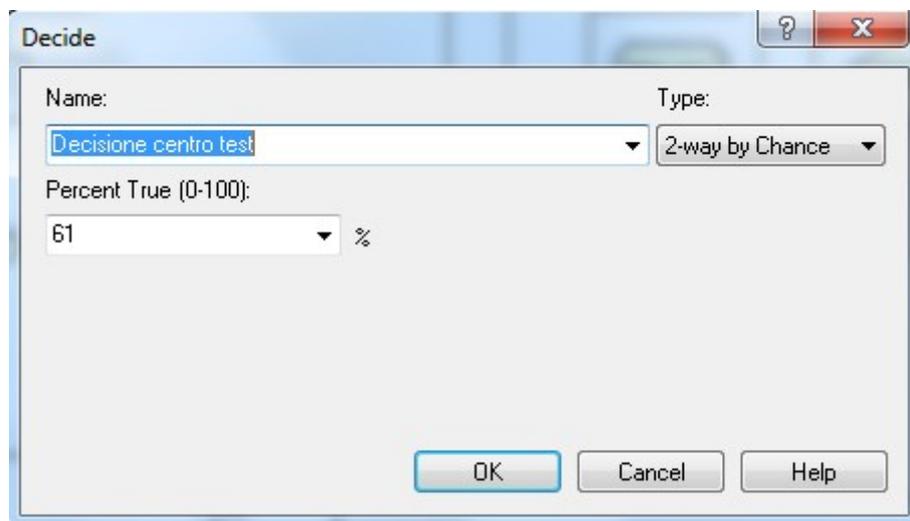


Figura 20: Decisione centro test

3.2 Settore 2: Centro di primo livello, Padova

Questo secondo settore comprende la zona di modello dedicata al centro di primo livello a Padova; comprende il test di primo livello e l'eventuale retest per risultato borderline. È composto da 10 moduli.

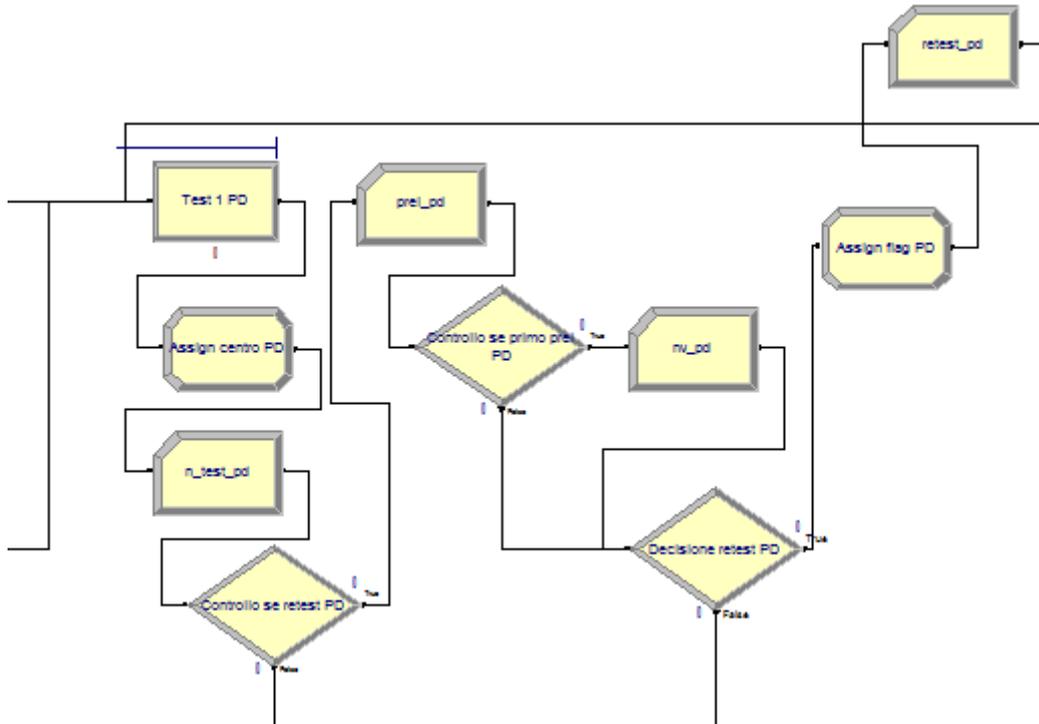


Figura 21: Settore 2

3.2.1 Primo test, Padova

Le entità giungono al primo modulo di questo settore dai due moduli “Decide” del settore 1, come precedentemente descritto. Questo primo modulo è il processo “Test 1 PD”. Esso è un processo del tipo “Seize Delay Release”; quindi necessita dell'uso di una risorsa e di una coda. La coda, nel diagramma, è realizzata attraverso la riga orizzontale sopra al modulo. La risorsa prende il nome di “Centro 1 livello PD”, di capacità 125 come già visto. Il tempo impiegato dal modulo a processare le entità è costante, pari a un giorno. Nella seguente figura è mostrato il pannello delle risorse; si può notare anche la presenza delle risorse speculari del centro di primo livello a Verona e di quello di secondo livello, costruite nello stesso modo, con solo capacità differente.

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Centro 1 livello PD	Fixed Capacity	125	0.0	0.0	0.0		0 rows	✓
2	Centro 1 livello VR	Fixed Capacity	85	0.0	0.0	0.0		0 rows	✓
3	Centro secondo livello	Fixed Capacity	10	0.0	0.0	0.0		0 rows	✓

Figura 22: Le risorse nel modello

Per quanto riguarda la coda, in Arena esistono 4 diversi tipi di code: FIFO (First In, First Out, il primo elemento che entra è anche il primo elemento che esce), LIFO (Last In, First Out, l'ultimo elemento che entra è il primo che esce), Lowest Attribute Value (ha la precedenza l'entità con valore minore di un certo attributo), Highest Attribute Value (ha la precedenza l'entità con valore maggiore di un certo attributo). Tutte le code che saranno presenti in questo modello (relative ai due centri di primo livello e a quello di secondo) saranno del tipo Highest Attribute Value; il nome del processo e dell'attributo a cui si riferiscono è mostrato nella figura seguente.

Queue - Basic Process					
	Name	Type	Attribute Name	Shared	Report Statistics
1	Test 1 PD.Queue	Highest Attribute Value	flag	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Test 1 VR.Queue	Highest Attribute Value	flag	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Test II livello.Queue	Highest Attribute Value	prior	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 23: Le code nel processo

Infine, nella figura seguente è mostrato il processo "Test 1 PD".

Figura 24: Test 1 PD

Successivamente, le entità attraversano il modulo “Assign centro PD” e “n_test_pd”. Nel primo, nel modo precedentemente descritto, viene assegnato il valore 1 all'attributo “centro”; il secondo è un contatore, per verificare il numero di test effettivamente eseguiti a Padova. Quindi si ha una verifica se il prelievo cui corrisponde l'entità sia già stato testato o meno; si tratta di un modulo “Decide” simile ai precedenti. Il controllo è dato dall'attributo “flag”: se essa ha valore pari a 0, non si tratta di un retest (parte true); se è diversa da 0, il prelievo è già stato testato due volte, e quindi si va direttamente ai risultati (parte false): infatti uno stesso prelievo non può essere testato più di due volte; questa parte va a finire in un altro settore. Seguendo la parte relativa al primo test, c'è un altro contatore, “prel_pd”, che conta il numero di prelievi diversi effettivamente controllati nel centro di Padova. Infine c'è un altro controllo, per verificare se si tratta di primo prelievo o secondo prelievo, attraverso sempre l'attributo “prel”. In caso di primo prelievo, si ha un altro contatore, “nv_pd”, per verificare il numero di entità diverse che sono state processate a Padova.

La differenza tra i tre contatori è che il primo considera il numero di test eseguiti nel complesso (i prelievi relativi ai retest vengono quindi contati due volte, come quelli dovuti a secondo prelievo), il secondo considera il numero di prelievi differenti (quindi due prelievi diversi ma riferiti alla stessa entità sono contati due volte), il terzo considera il numero di entità diverse che transitano per Padova.

3.2.2 Secondo test, Padova

Come già scritto, ogni prelievo ha il 10% di possibilità di avere risultati borderline. In caso ciò avvenga, verrà effettuato un secondo test, sempre sul medesimo prelievo nel medesimo centro. Questa possibilità è data dal modulo “Decisione retest PD”, di tipo 2-way by chance.

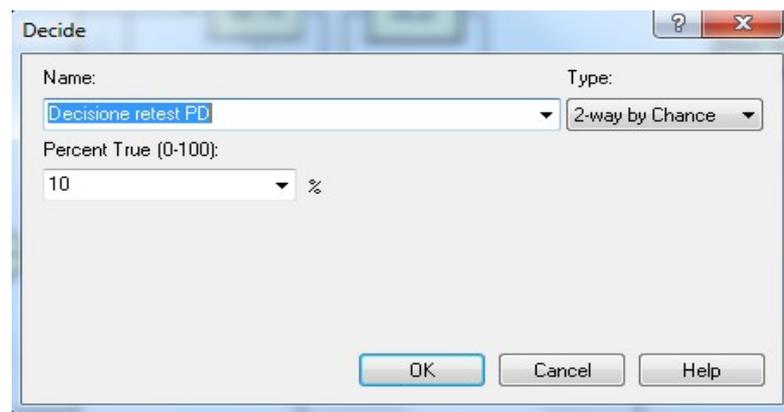


Figura 25: Decisione retest PD

In caso affermativo, quindi con obbligo del secondo test, l'attributo “flag” indicativo del test viene settato a 1 con il modulo “Assign flag PD”, e si verifica il numero delle entità testate una seconda volta nel contatore “retest_pd”. Le entità così processate ritornano al test di primo livello, dove avranno la precedenza.

3.3 Settore 3: Centro di primo livello, Verona

Questa parte di modello è del tutto speculare alla precedente: infatti, le azioni che vengono compiute nei due centri sono perfettamente le stesse.

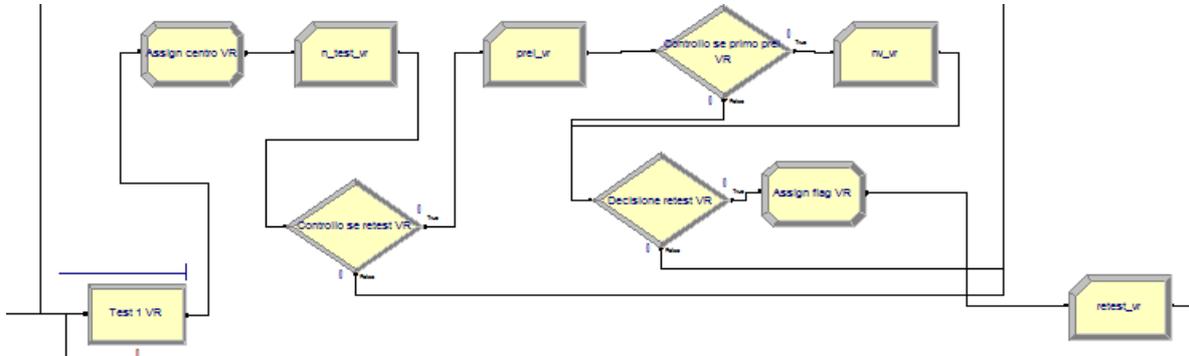


Figura 26: Settore 3

Ciò che cambia è che qui si giunge dai valori “false” delle due decisioni precedenti, e che il valore dati alla capacità del centro di primo livello è 85, mentre all'attributo “centro” è 2. Per il resto, i contatori, le decisioni e il processo avvengono nello stesso modo visto per il centro di primo livello di Padova.

3.4 Settore 4: Risultati dei test

Se il prelievo risulta essere già stato testato due volte, o non necessita di un secondo test, si giunge al risultato dei test. Questo è espresso nel settore 4, in cui si mostra anche la prima fase del processo riguardante i prelievi positivi.

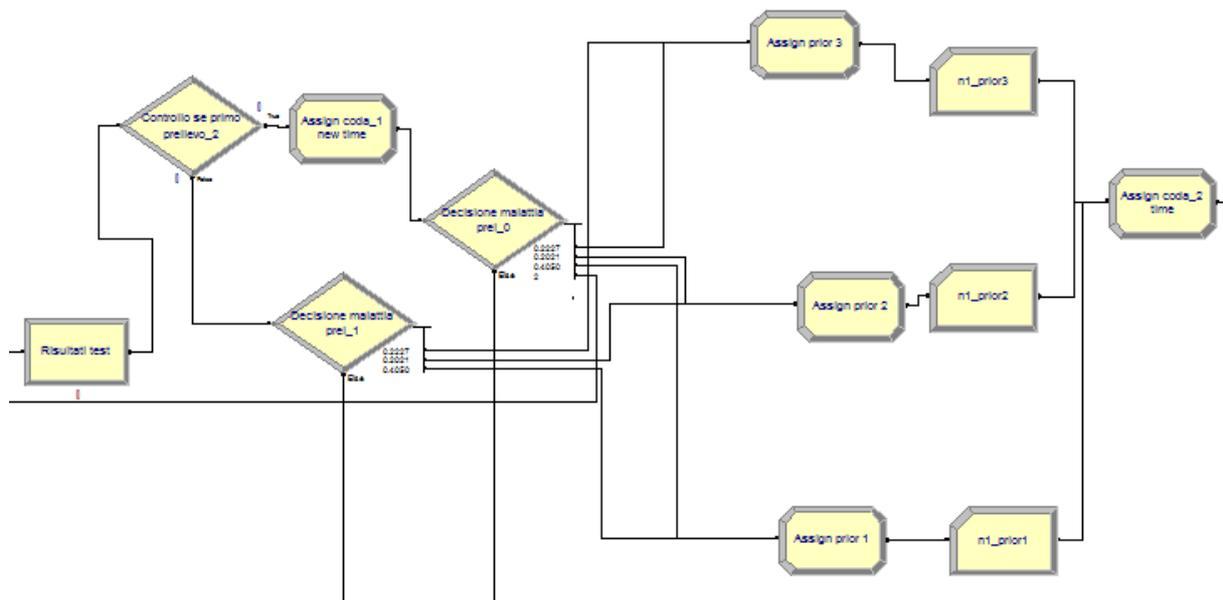


Figura 27: Settore 4

Il primo processo è “Risultati test”. Esso è in realtà un processo fittizio, in quanto è di tipo Delay con tempo di processo costante pari a 0. La sua utilità è di rendere più chiaro il tutto, in quanto per la sua posizione divide in due parti il modello: una prima parte più di processo vero e proprio, e una seconda parte più di statistiche utili all'analisi.

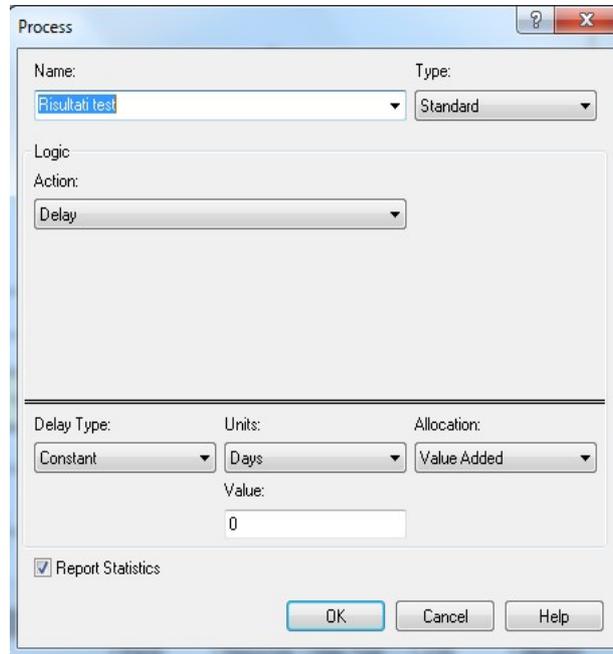


Figura 28: Risultati test

A seguire, c'è un nuovo controllo sul fatto che si tratti o meno di primo prelievo, sempre grazie all'attributo “prel” (modulo “Controllo se primo prelievo_2”). Infatti, se è già stato fatto un secondo prelievo, va a mancare quel 2% di possibilità che si avrebbe nel caso contrario. Perciò si dovranno utilizzare due moduli “Decide” differenti per i due casi. Nel caso si tratti di primo prelievo, si va poi a modificare il valore dell'attributo coda_1: esso viene infatti settato al valore TNOW-coda_1. Ciò significa che ora il suo valore diventa esattamente quello di percorrenza del modello dall'arrivo al centro di primo livello all'uscita da esso.

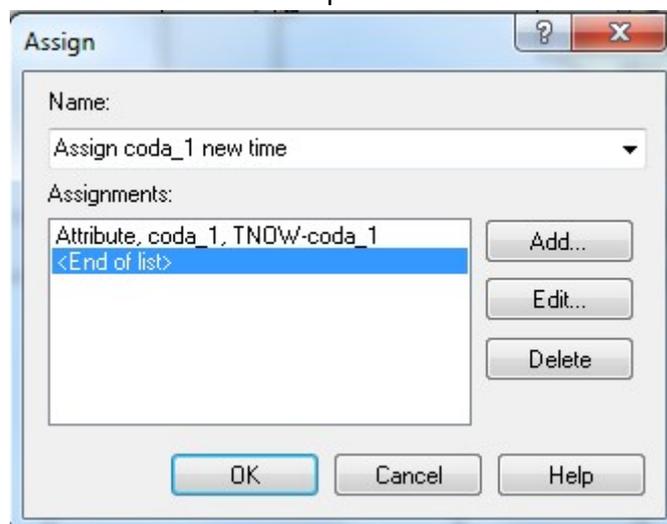


Figura 29: Assign coda_1 new time

Ora viene valutato se il prelievo risulta essere positivo o negativo, attraverso le probabilità già indicate. Si utilizzano i moduli “Decide” di tipo N-way by condition. Per ogni possibilità, nasce una freccia nel modello; in tutto si hanno 5 possibilità (positivo di priorità 1, positivo di priorità 2, positivo di priorità 3, negativo, nascita prematura) per i primi prelievi, 4 per i secondi (non si ha la possibilità di fare un ulteriore terzo prelievo). Una delle possibilità non è esplicitata con la sua percentuale nel modulo “Decide”; essa è infatti ottenuta dalla differenza tra il 100% e le altre percentuali, e viene indicata da “Else”, nella parte bassa del modulo. Si è deciso che tale parte sia riferita alla percentuale maggiore, e che quindi sia relativa ai prelievi risultati negativi.

Viene presentato lo schema relativo ai primi prelievi.

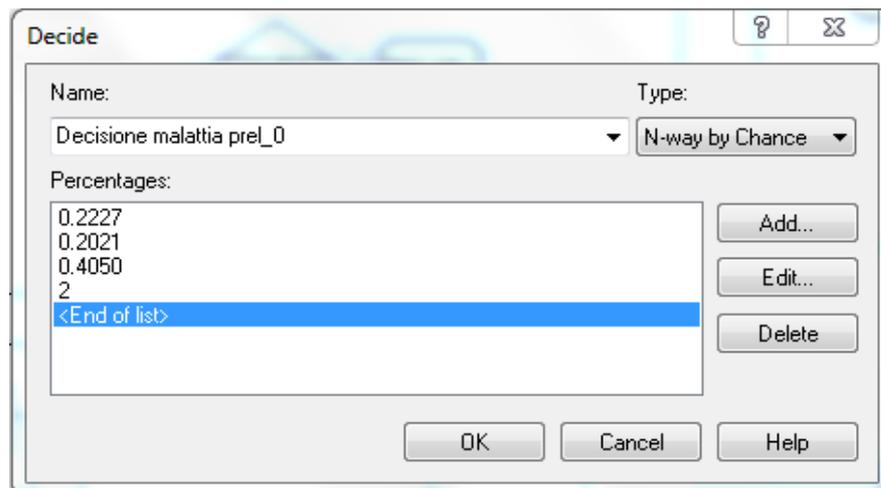


Figure 30 e 31: Decisione malattia prel_0

Il modulo “Decisione malattia prel_1” è uguale, a parte il fatto che non è presente il 2 nella lista delle percentuali.

A seconda del ramo che viene percorso dai prelievi risultanti positivi, si avrà una malattia di priorità uno, due, o tre; nei rispettivi moduli “Assign” viene dunque assegnato all'attributo “prior” il valore di priorità relativo: quindi 1, 2 o 3. Nei contatori seguenti viene segnato il numero di entità che risultano positive per ogni diversa priorità. Infine, ad ogni entità risultata positiva, è assegnato l'attributo coda_2, il cui valore diventa TNOW; questo attributo, come i precedenti relativi ai tempi, serve per individuare il tempo di percorrenza a partire dall'arrivo al centro di secondo livello, che sarà il modulo successivo.

3.5 Settore 5: Risultati negativi e secondo prelievo

In questo settore sono accorpati i prelievi risultati negativi e quelli per cui è necessario un secondo prelievo, anche se nel modello sono situati su porzioni di spazio completamente diverse.

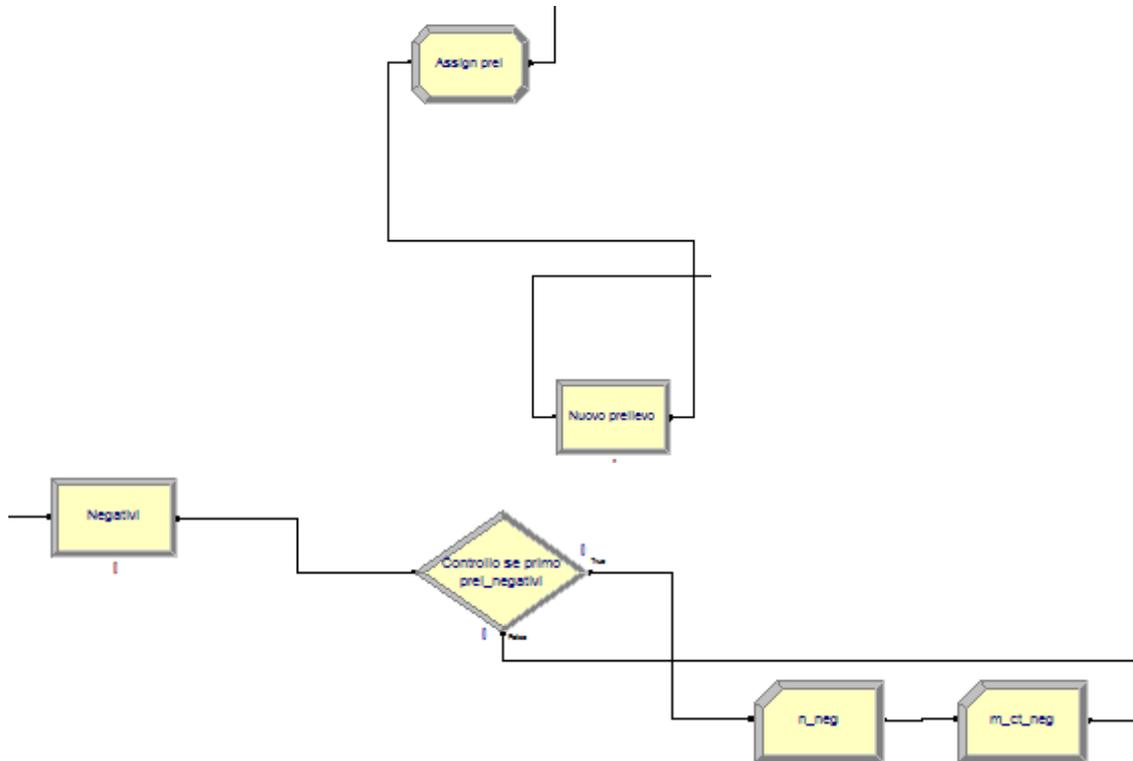


Figure 32 e 33: Settore 5

Nei processi “Nuovo prelievo” e “Negativi” si giunge dai moduli “Decide” del settore precedente, come spiegato. Per quanto riguarda la parte relativa ai nuovi prelievi, il processo è di tipo “Delay”, con durata costante pari a 15 giorni; all'uscita da questo modulo, all'attributo “prel” viene dato valore 1 nel modulo “Assign prel”. Dopodichè si torna al processo di spedizione visto nel settore 1.

Per quanto riguarda i prelievi risultanti negativi, il processo “Negativi” è un altro processo fittizio, creato per migliorare la comprensibilità del modello, di tipo “Delay” con tempo costante pari a 0. Quindi si va a verificare ancora una volta se il prelievo è un primo prelievo o meno; in caso affermativo, si procede a contare il numero di entità nel modulo “n_neg”, che è il classico modulo di tipo contatore. Al contrario, il modulo “m_ct_neg” è un modulo “Record” di tipo “Time Interval”. In esso il programma va a valutare il valore della differenza tra TNOW e l'attributo inserito nel modulo, in questo caso coda_tot, e ne calcola la media, che verrà riportata nei risultati al termine della corsa del modello. Questa media è la media dei tempi impiegati dai prelievi risultati negativi per attraversare tutto il modello (infatti il modulo che segue “m_ct_neg” è il modulo finale), quindi per ottenere un risultato a partire dal tempo della nascita.

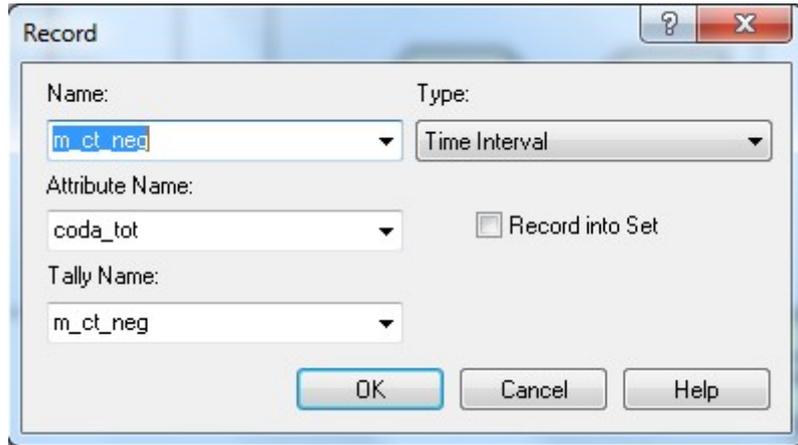


Figura 34: Modulo m_ct_neg

3.6 Settore 6: Test di secondo livello e raccolta dati

In questa parte si ha l'ultimo dei processi richiesti dal modello, e una serie di moduli di tipo "Record" per immagazzinare i dati su cui svolgere poi l'analisi.

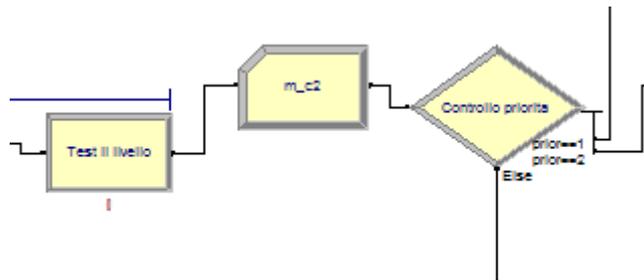


Figura 35: Settore 6 – parte 1

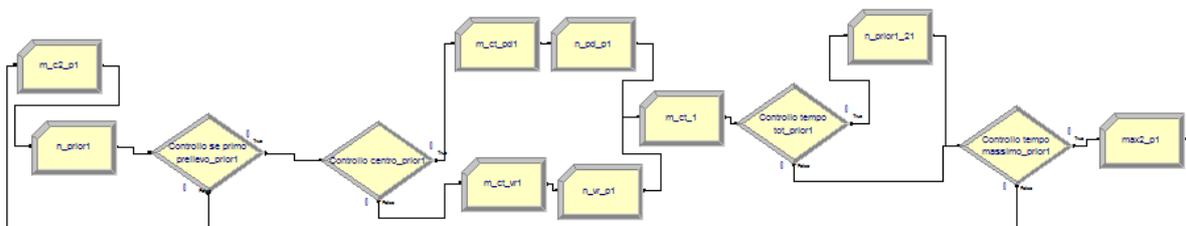


Figura 36: Settore 6 – priorit  1

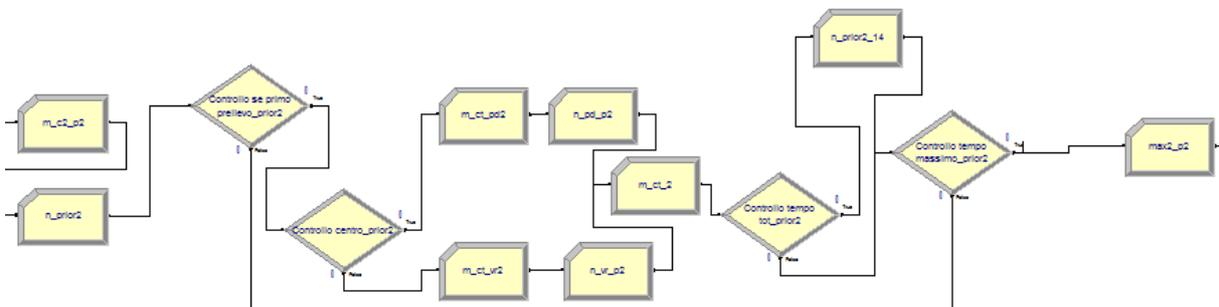


Figura 37: Settore 6 – priorit  2

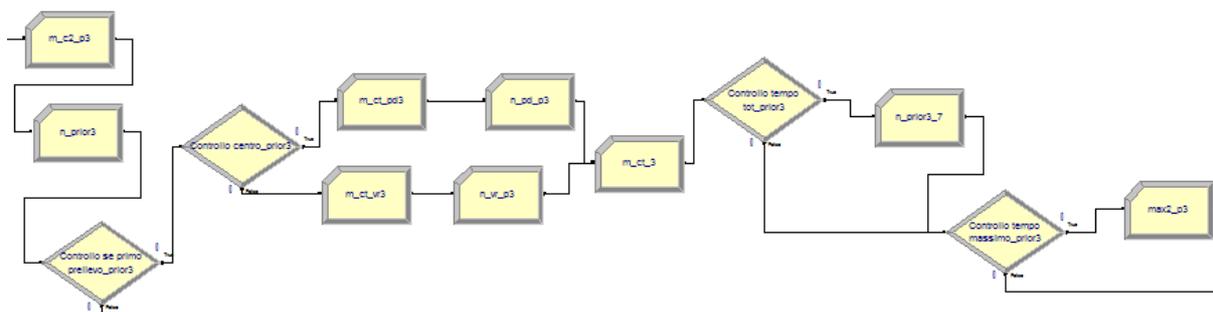


Figura 38: Settore 6 – priorità 3

3.6.1 Centro di secondo livello

Il processo riguardante il centro di secondo livello è il medesimo dei precedenti per i centri di primo livello: processo di tipo “Seize Delay Release”, con quindi una risorsa (il centro di secondo livello, capacità 10) e una coda (sempre di tipo Highest Attribute Value), il cui attributo di riferimento è però “prior” al posto di “flag”. Il tempo impiegato è costante, pari a 0,625 giorni, ovvero 15 ore. Il processo è seguito dal modulo “m_c2”, del tutto simile al modulo “m_ct_neg” visto in precedenza; questa volta, l'attributo interessato è coda_2. Dunque questo modulo va a calcolare la media dei tempi impiegati a ottenere il risultato del test di secondo livello. A seguire, c'è il modulo “Controllo prioritá”, modulo “Decide” di tipo N-way by condition; esso va a dividere le entità che processa a seconda della loro priorità. Da questo modulo nascono tre rami del programma perfettamente simmetrici, in dipendenza della priorità. Ne verrà analizzato solo uno, essendo gli altri identici; l'unico modulo che tra essi differisce in parte verrà segnalato.

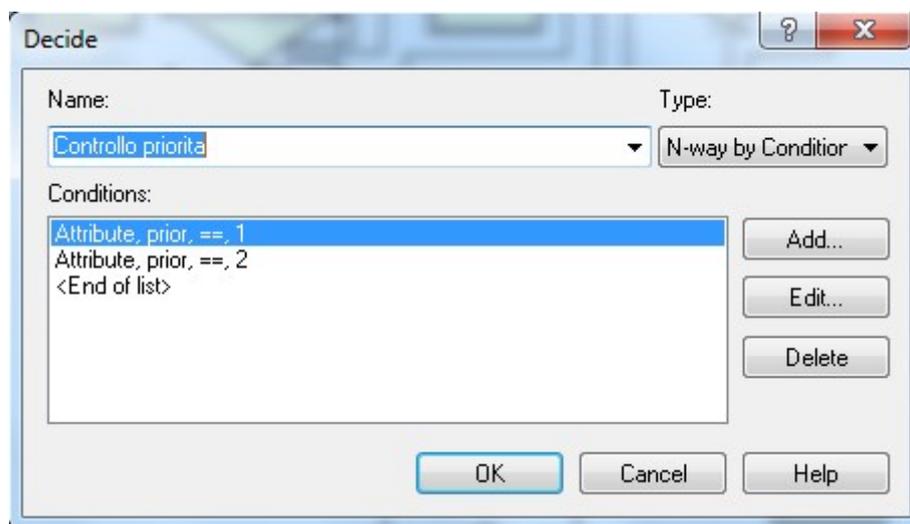


Figura 39: Controllo prioritá

3.6.2 Raccolta dati

Si analizza la parte relativa alla priorità 1. Il primo modulo, "m_c2_p1", è di tipo "Time Interval", con attributo di competenza coda_2: raccoglie la media del tempo occorso per attraversare il centro di secondo livello relativamente alle entità di priorità 1. Il seguente, "n_prior_1", è un contatore e segna il numero di entità di priorità uno che attraversano il modello. Quindi c'è, ancora una volta, un controllo per valutare se il prelievo analizzato è primo o secondo prelievo; se è secondo, si esce dal ramo per andare nel modulo "n", che verrà mostrato in seguito. Se invece è primo, si prosegue verso un altro controllo, questa volta del centro di primo livello di provenienza; da qui partono due rami simmetrici, uno relativo al centro di Padova e uno al centro di Verona. In essi sono situati due moduli: "m_ct_pd_1 (o vr_1)" raccoglie la media dei tempi intercorsi tra l'inizio del modello e il momento corrente relativamente alle entità transitate per Padova (o Verona) di priorità 1; il secondo, "n_pd_p1 (o vr_p1)" conta questo numero di entità. I due rami si riuniscono nel modulo "m_ct_1", media degli stessi tempi dei moduli precedenti ma relativo a tutte le entità di priorità 1. A seguire, è presente il modulo "Controllo tempo tot_prior1", di tipo "Decide", 2-way by condition, ma guidato da un'espressione. Se l'espressione è soddisfatta, si andrà dalla parte relativa a "true"; al contrario, dalla parte "false". In questo caso, l'espressione in oggetto è $TNOW-coda_tot > 21$; ciò serve a verificare se alcune entità hanno accumulato un ritardo non accettabile per la priorità 1, cioè maggiore a tre settimane. In caso affermativo, nel modulo "n_prior_21" si tiene il conto del loro numero.

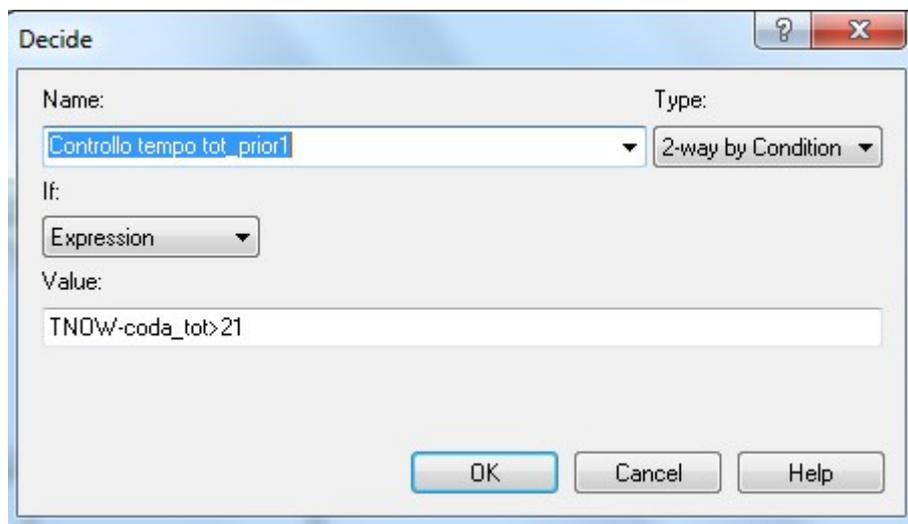


Figura 40: Controllo tempo tot_prior1

Questa è l'unica parte che varia leggermente tra i tre rami: infatti, per la priorità 2 l'espressione diventa $TNOW-coda_tot > 14$, mentre per la priorità 3 $TNOW-coda_tot > 7$, in quanto le entità con priorità due vanno processate entro 2 settimane, quelle di priorità 3 entro una settimana.

Infine, si va a ricercare l'entità che impiega il tempo massimo a venire processata. Questo avviene nel modulo "Controllo tempo massimo_prior1", che dipende da un'espressione, che è $TNOW-coda_2 > max2_p1$. "max2_p1" è il modulo successivo, sempre di tipo "Time Interval" con attributo di riferimento coda_2. Ciò vuol dire che, se un'entità impiega più del tempo medio impiegato finora dalle altre (ovviamente, questo tempo aumenta man mano col passare del modello), questo suo tempo viene registrato nel modulo "max2_p1". Alla fine però, di questo modulo non interesserà la media, ma il valore massimo, corrispondente appunto all'entità che ha impiegato più tempo.

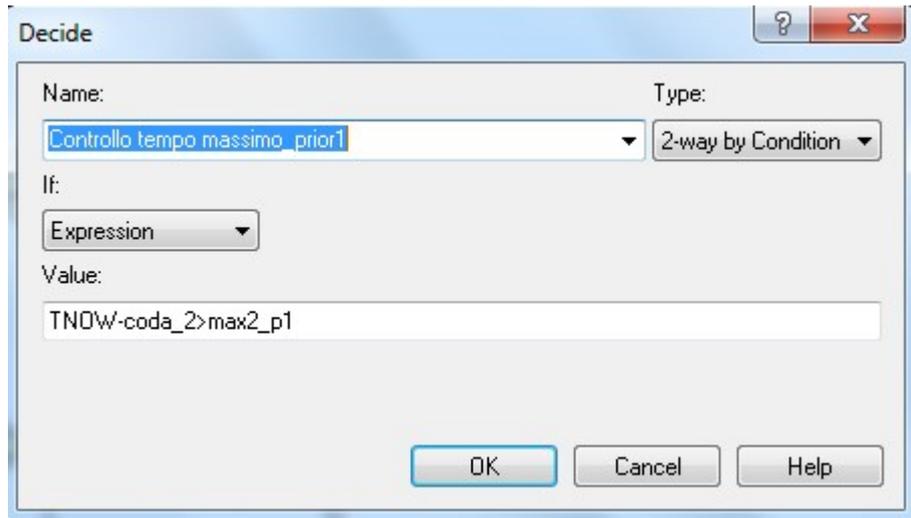


Figura 41: Controllo tempo massimo_prior1

Le entità uscenti dai moduli "max2_p1", "Controllo tempo massimo_prior1" (lato "false") e "Controllo se primo prelievo_prior1", unite a quelle delle altre due priorità, entrano tutte nel modulo "n".

3.7 Settore 7: Uscita

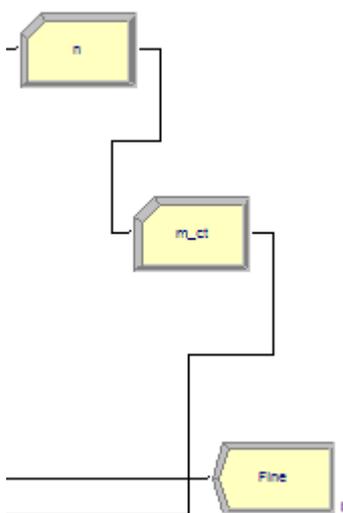


Figura 42: Settore 7

Le entità che risultano positive arrivano tutte, come detto, al modulo "n", un contatore, che tiene appunto conto di esse. Quindi nel modulo "m_ct" si tiene conto della media totale dei tempi di attraversamento, di tutte le entità risultate positive indipendentemente dalla priorità. Infine, riunendosi con le entità provenienti dal ramo dei risultati negativi, escono tutte dal modello nel modulo "Fine", di tipo "Dispose".

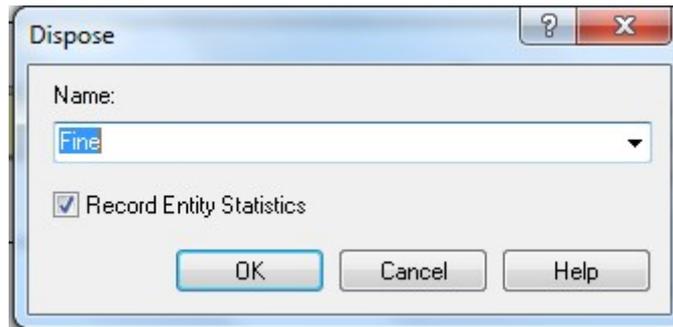


Figura 43: Fine

Infine, nel menù Run → Setup, nella finestra Replication Parameters, vanno indicati il Warm-up Period di 365 giorni e la Replication Length (durata della simulazione) di altrettanti 365 giorni.

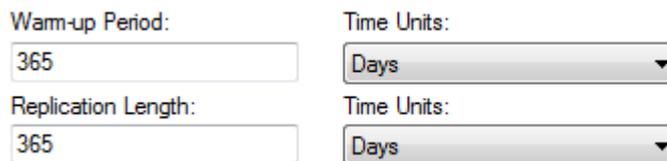


Figura 44: definizione dei tempi del modello

Conclusioni

Si è mostrato come realizzare un modello di simulazione per lo screening neonatale su software Arena, utilizzando solo i processi base presenti nel software. È poi possibile fare su di esso esperimenti modificandone alcuni dati (come per esempio la capacità dei vari centri). Infine, si possono analizzare tutti i dati raccolti e valutare tra le diverse alternative quale risulti essere la più soddisfacente. Tuttavia, questa fase esula dal contesto di questo lavoro, basato sulla costruzione del modello.

BIBLIOGRAFIA

Romanin Jacur G., “La simulazione”;

Arena User's Guide, by Rockwell Automation;

Romanin Jacur G., Da-Frè M., Mazzuccato M., Minichiello C., Visonà-Dalla Pozza L.,
“Neonatal Expanded Screening for Inherited Metabolic Diseases: Planning by Simulation”.

RINGRAZIAMENTI

Per primi, ringrazio i miei genitori, che mi sono sempre stati vicini e mi hanno sempre sostenuto in questi tre anni, dandomi forza nei momenti di difficoltà e non facendomi mai adagiare su quanto già ottenuto.

Ringrazio mio fratello, che mi ha aiutato quando ne avevo bisogno, e che so essere orgoglioso di me come io lo sono di lui.

Ringrazio il professor Romanin Jacur, per la sua disponibilità nel risolvere i dubbi e i problemi riscontrati durante la realizzazione della tesi.

Ringrazio Giampi, che più di ogni altro in questi anni mi ha sopportato, aiutato, incoraggiato e quant'altro, tra lezioni, viaggi in treno e serate varie.

E infine, ringrazio "I Ragazzi", Alice, Davide, Giova, Giulia, Ilenia, Jacopo, Luca, Marco, Michi, Pippo, Tia: senza di loro l'università non sarebbe stata la stessa cosa.