

## Capitolo 7

### Dal 118 reale al 118 virtuale

Fino ad ora si è parlato degli aspetti più generali che riguardano la sanità, le organizzazioni di pronto intervento sanitario, la complessità e la simulazione, trattando solo in modo marginale gli aspetti più caratteristici della gestione di una centrale operativa e soprattutto non trattando affatto le problematiche inerenti la modellizzazione di una centrale operativa del 118.

È facile immaginare che per poter costruire un modello di qualcosa si deve essere in grado di descrivere, in modo dettagliato e fedele, l'oggetto che si vuole modellizzare. È quindi condizione necessaria per costruire un modello conoscere in modo minuzioso l'oggetto della trattazione, se l'oggetto di studio non si conosce allora si ha la necessità che qualcuno ce lo descriva in modo preciso.

Quando abbiamo<sup>1</sup> deciso di creare un modello di simulazione di una centrale operativa del 118 non avevamo idea di come funzionasse questa realtà se non l'idea che ogni cittadino ha riguardo al funzionamento del servizio di prima emergenza sanitaria. Grazie alla disponibilità dei dirigenti della centrale operativa di Grugliasco, e grazie soprattutto al tempo che ci è

---

<sup>1</sup> L'uso del plurale è doveroso dato che in questa "avventura" non sono stata sola ma sono stata "accompagnata" da altri studenti che hanno deciso di fare una tesi sullo stesso argomento e con me hanno costruito il modello sto per descrivere.

stato dedicato dagli stessi siamo stati in grado di acquisire le conoscenze necessarie per iniziare a creare quello che abbiamo soprannominato il “118 virtuale”.

Per descrivere il percorso che ci ha condotto fino alla creazione di un modello che descrivesse nel modo più verosimile possibile la realtà del servizio di pronto intervento sanitario conviene riportare i passi successivi che ho condotto con i miei compagni fino a giungere alla costruzione del 118 virtuale.

## 7.1 La gestione della centrale operativa

La gestione della centrale operativa di Grugliasco si configura, graficamente, nella seguente maniera.

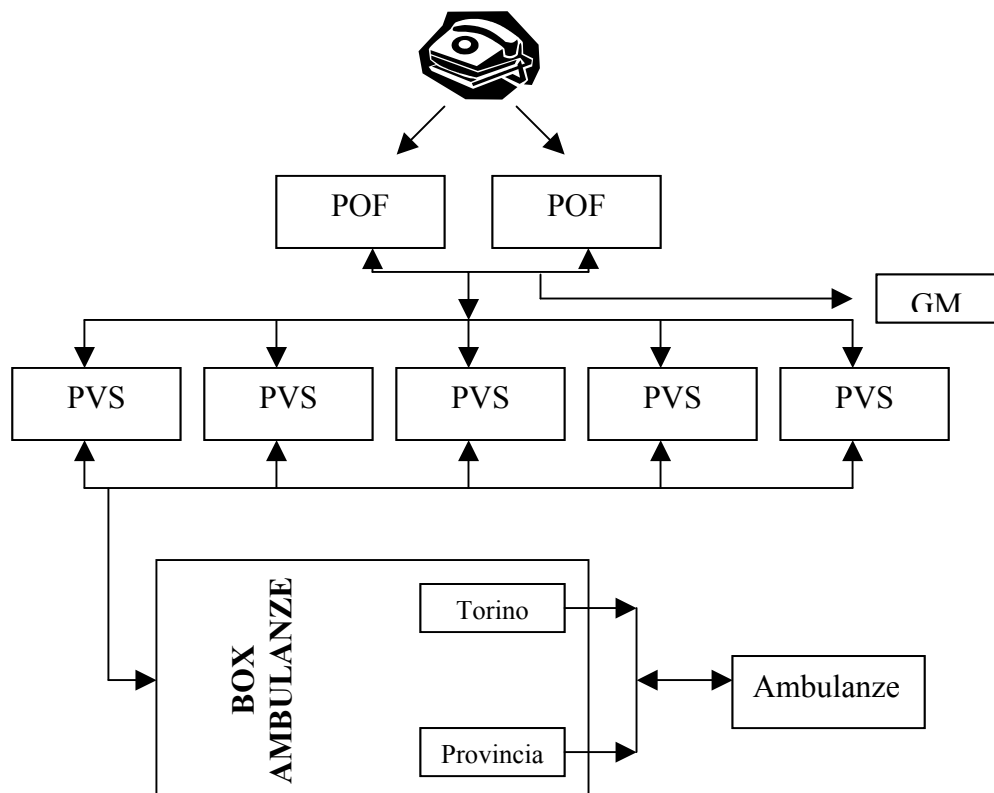


Figura 1- 118 grafico

Come si vede dalla *Figura 1* la chiamata arriva al Posto Operatore Filtro (**POF**) il quale ha come compito quello di smistare le chiamate destinate alla guardia medica (che si trova fisicamente all'interno della struttura della centrale operativa anche se si tratta di una realtà distinta ed autonoma rispetto a questa e che presto verrà trasferita) da quelle indirizzate al 118, inoltrando queste ultime alle Postazioni di Valutazione Sanitaria (**PVS**). Le mansioni affidate agli operatori del **POF** sono di natura tecnica e per essere svolte non c'è bisogno che questi operatori abbiano competenze mediche, è sufficiente che abbiano competenze tecniche.

Dopo essere passata dal **POF** la chiamata arriva alle **PVS** nelle quali si effettua una valutazione dello stato del paziente e dove si compila una scheda che viene inoltrata al **Box Ambulanze**. Gli infermieri che si trovano alla postazione di valutazione possono prestare i primi interventi, cercando di far eseguire alcune semplici operazioni di medicina di base ai presenti sul luogo dell'incidente prima dell'arrivo dell'ambulanza. Gli operatori che si occupano di questa fase del pronto intervento devono necessariamente essere infermieri professionali.

Le **PVS** generano degli eventi ai quali sono assegnati dei codici di gravità e che vengono tempestivamente trasmessi al box ambulanze, nel quale si trovano due operatori tecnici, uno di questi si occupa di gestire gli interventi effettuati nella provincia di Torino mentre l'altro operatore si occupa di gestire gli interventi che vengono effettuati sulla città di Torino. Questi operatori svolgono le azioni necessarie a gestire le comunicazioni tra ambulanza e centrale operativa. Ad affiancare questi operatori si trova un infermiere professionale che interviene nel caso in cui vadano prese importanti decisioni sanitarie quali, per esempio, la scelta dell'ospedale, o l'invio di un altro mezzo nel caso vi sia stata una sottostima della gravità dell'intervento. Dopo la scelta del mezzo l'evento diviene una “**missione operativa**”, ed ogni missione si compone di otto step:

- *ricerca mezzo;*
- *allarme al mezzo;*
- *partenza;*
- *sul posto;*
- *paziente a bordo, in partenza;*
- *arrivo all'ospedale;*
- *di nuovo operativi;*
- *chiusura servizio.*

Per un'analisi più approfondita del funzionamento e del significato delle singole voci sopra riportate rimando al *Capitolo 2*.

Il 118 ha a disposizione tre tipi di mezzi di soccorso: **MSA** (mezzo di soccorso avanzato) sul quale sono presenti 1 medico, 1 infermiere e 2 soccorritori volontari, **MSAB** (mezzo di soccorso avanzato di base) sul quale sono presenti 1 infermiere e 2 soccorritori volontari e l'**MSB** (mezzo di soccorso di base) sul quale sono presenti 2 soccorritori volontari.

La prima impressione che abbiamo avuto quando ci è stato descritto il funzionamento della centrale operativa è che sembrano esserci all'interno della centrale operativa almeno due strozzature. È indubbio infatti che i carichi di lavoro più grossi sono quelli che gravano sui Posti Operatore Filtro e sui Box Ambulanze. Vedremo se il modello che riproduce una giornata confermerà o smentirà quanto detto.

A questo punto il modello ha incominciato a prendere forma nella nostra testa ma quanto ci è stato spiegato non era ancora sufficiente per costruire realmente il *118 virtuale*, infatti avevamo bisogno di analizzare i dati e decidere come agire.

## 7.2 I dati

In seguito alle visite da noi compiute presso la sede della centrale operativa di Grugliasco ed in seguito alla nostra richiesta di poter visionare i dati ci è stato fornito un database, inerente all'anno 2002, che raccoglie i dati di tutti gli interventi effettuati dalle ambulanze in quell'anno.

La richiesta di avere dei dati reali nasce dalla necessità di costruire il modello in modo che riproduca la realtà, e quindi facendo in modo che il modello stesso risulti attinente con quanto accaduto realmente all'interno della centrale operativa avendo così un riscontro con la realtà. Naturalmente se il modello costruito risulterà veritiero è allora verosimile immaginare di inserire all'interno del modello alcune modifiche all'attuale struttura della centrale operativa, analizzare i risultati delle modifiche e poi pensare se modificare realmente la centrale operativa.

### 7.2.1 Il Database

Il database che si è stato fornito è creato tramite Microsoft Access ed è composto da diversi campi all'interno dei quali sono inserite le voci che li distinguono.

I campi che troviamo all'interno della banca dati del 118 sono i seguenti<sup>2</sup>:

- ID scheda;
- ID scheda collegata;
- data scheda, ora scheda;
- chiamante località, chiamante indirizzo, chiamante numero;
- tipo di risposta;

---

<sup>2</sup> In questo elenco raggrupperò in un unico punto alcune voci che pur trovandosi in campi diversi sono simili.

- codice mezzo;
- tipo mezzo;
- disponibilità;
- postazione;
- associazione;
- ora allarme, ora partenza, ora arrivo;
- ora partenza per l'ospedale, ora arrivo all'ospedale, ora operativo;
- codice di criticità rientro, codice patologia rientro, codice luogo rientro ;
- codice di rientro;
- esito;
- descrizione non trasportato , descrizione non effettuato;
- codice mezzo trasporto;
- ospedale di trasporto.

Descriverò singolarmente le voci che sono presenti all'interno del database.

### ID scheda

All'interno di questo campo è inserito il numero identificativo della telefonata arrivata al 118. I numeri che compaiono in questo campo sono inseriti in ordine crescente in modo da poter avere una contabilità immediata del "traffico" e sono anche univoci, in modo da poter distinguere tutti i casi dal solo ID. Il numero ID viene azzerato ad ogni primo dell'anno.

### Collegata

In questa colonna sono inseriti i numeri di quei casi particolari che, a giudizio dell'operatore del PVS o dell'infermiere che affianca gli operatori del Box Ambulanze, richiedono l'intervento di più ambulanze.

Quindi ciò che capita, qualora il caso sia particolarmente grave, tale da richiedere l'intervento di più mezzi di soccorso sul luogo dell'incidente, è la creazione di più schede collegate tra di loro.

Il legame tra le schede viene registrato sul database tramite il riempimento del campo "Collegata". All'interno di questo campo viene inserito il numero della prima scheda che genera l'evento.

Esemplifichiamo attingendo un caso dal database.

IDScheda	Collegata	DataScheda	OraScheda
000041	000041	01/01/02	01:25
000045	000041	01/01/02	01:25
000046	000041	01/01/02	01:25

**Tabella 1- Caso schede collegate**

Come si può notare dalla *tabella 1* la scheda 000041 ha generato un evento che richiede l'intervento di più di un'ambulanza, quindi gli operatori della centrale operativa hanno generato due schede successive collegate alla prima scheda generatrice. Il collegamento è visibile perché nella colonna chiamata "Collegata" compare il numero della scheda generatrice del caso.

Va sottolineato come la creazione di queste schede collegate non sempre derivi da chiamate diverse e quindi non sempre queste schede creano lavoro aggiuntivo al POF ed al PVS, mentre sicuramente vi è un aggravio di lavoro per ciò che riguarda il Box ambulanze che si trova a dover gestire più ambulanze per un unico caso.

### Data scheda / Ora scheda

Sono i campi in cui vengono inseriti, in modo intuitivo, l'ora della scheda, che corrisponde all'ora in cui la chiamata giunge al PVS, e la data della scheda, che si riferisce alla data in cui è avvenuto l'evento negativo.

### Chiamante località, Chiamante indirizzo, Chiamante numero

Qui sono inseriti i dati geografici che servono per indicare dove si trova la persona da trasportare e quindi sono dati strettamente necessari nella gestione della centrale operativa perché permettono all'ambulanza di giungere sul posto che altrimenti non sarebbe raggiungibile. Questi dati in una seconda elaborazione del modello saranno molto utili.

### Tipo di risposta

In questo campo sono inserite le risposte che la centrale operativa fornisce agli utenti.

Infatti in alcuni casi la centrale operativa non invia alcun mezzo di soccorso ma fornisce semplicemente una risposta telefonica al chiamante.

Questa situazione si verifica quando gli infermieri professionali, che svolgono il loro compito operando sul PVS, non ritengono necessario l'invio di un mezzo perché il caso è gestibile tramite alcuni interventi che possono essere guidati telefonicamente o semplicemente perché ritengono che il caso non sia grave e quindi non richiede alcun intervento.

### Codice mezzo

È il codice di riconoscimento dell'ambulanza, infatti ad ogni mezzo di soccorso corrisponde una sigla<sup>3</sup> formata dalla sigla TO e seguita da un numero univoco per ogni ambulanza. Particolarità di questo tipo di numerazione è che i mezzi di soccorso avanzati ed avanzati di base sono identificati con un numero che si conclude sempre con un 5.

---

<sup>3</sup> In appendice sono elencati tutti i mezzi di soccorso utilizzati su Torino e provincia

### Tipo mezzo

Si riferisce alla distinzione tra Mezzi di Soccorso di Base, Mezzi di Soccorso di Base Avanzati e Mezzi di Soccorso Avanzati ed indica quale tipo di mezzo è intervenuto sul luogo dell'incidente.

### Disponibilità

Riguarda il tipo di convenzione che il 118 ha con le associazioni che sono le proprietarie delle ambulanze che può utilizzare per il suo servizio di soccorso.

Gli accordi possibili tra 118 ed associazioni sono solo di due tipi e cioè l'ambulanza può essere disponibile ventiquattro ore su ventiquattro, ed, in questo caso, la sigla che si troverà sul database è *H24*. Se l'ambulanza può essere disponibile solo in certe fasce orarie ed in certi giorni, in questo caso, sul database si troverà la dicitura *estemporanea*

### Postazione

Indica dove l'ambulanza è sita. Questo tipo di informazione è importantissima per mandare il mezzo di soccorso disponibile più vicino al luogo dell'incidente. A questo riguardo il sistema informativo del 118 ha inserito al suo interno un foglio excel che fornisce l'indicazione delle competenze per ogni singolo mezzo utilizzabile e disponibile nel momento in cui lo si richiede, così che lo stesso sistema potrà suggerire il mezzo disponibile più vicino al luogo dell'incidente.

### Associazione

Segnala a quale associazione appartiene il mezzo di soccorso utilizzato per far fronte all'evento.

### Ora allarme, Ora partenza, Ora arrivo

Sono gli orari che indicano quando il mezzo è stato allertato, quando questo è partito per andare sul luogo dell'incidente e quando è arrivato sul luogo stesso.

### Ora partenza ospedale, Ora arrivo ospedale

Indicano quando il mezzo di soccorso giunto sul luogo dell'incidente riesce a ripartire per l'ospedale dopo aver compiuto le normali operazioni di stabilizzazione del malato, e quando arriva all'ospedale.

### Ora operativo

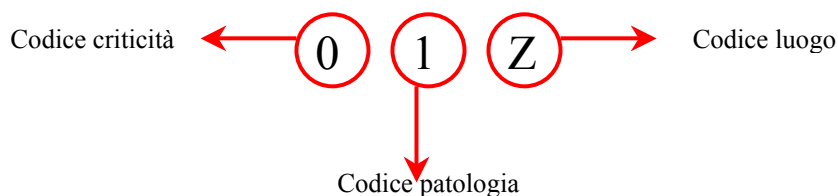
Mostra l'orario in cui l'ambulanza è nuovamente operativa e quindi il momento in cui questa può essere richiamata per altri interventi.

### Cod. criticità rientro, Cod. patologia rientro, Cod. luogo

Indicano quanto è grave il malato, il tipo di patologia che lo affligge ed il luogo in cui si è manifestata la patologia.

### Cod. rientro

È riassuntivo di tutti i codici<sup>4</sup> precedenti ed è strutturato come nell'esempio



**Figura 2- Struttura del codice di rientro**

---

<sup>4</sup> In appendice si trovano tutti i codici utilizzati durante lo svolgimento dell'attività di soccorso.

### Esito

Indica se il trasporto è stato effettuato o meno, le motivazioni di un mancato trasporto sono indicate nelle due voci successive.

### Descrizione non trasportato, Descrizione non effettuato

Illustra il motivo per cui il trasporto non è stato effettuato.

### Cod. mezzo trasporto

Indica il mezzo che ha effettivamente effettuato il trasporto. Questa voce molto spesso è uguale alla voce precedente che indica, come questo campo, il codice del mezzo inviato. Le uniche dissonanze si hanno nel caso in cui siano state inviate sul luogo dell'incidente più mezzi. Infatti in questo caso può capitare che, nonostante siano stati allertati diversi mezzi, solo uno o solo alcuni di essi effettuino il trasporto all'ospedale dei malati mentre gli altri mezzi tornano indietro senza effettuare alcun trasporto.

### Ospedale trasportato

Indica l'ospedale in cui è stato trasportato il paziente. Questa voce potrebbe essere utile se si volessero fare degli studi sulla razionalizzazione dei servizi ospedalieri sul territorio provinciale. Lo scopo dello studio sarebbe verificare se l'obiettivo della razionalizzazione dell'ospedalizzazione sul territorio, che è solo uno degli obiettivi che lo Stato e le Regioni si erano proposti con l'introduzione del servizio di pronto intervento sanitario, è stato rispettato.

### 7.2.2 L'analisi dei dati

Quando siamo entrati in possesso del database abbiamo fatto un'iniziale analisi sui dati complessivi per capire l'utilizzo dei mezzi sul territorio e per evidenziare eventuali disparità e quindi possibili margini di miglioramento attraverso l'ottimizzazione dell'impiego

Per ciò che riguarda Torino è da subito emerso che durante l'anno 2002 vi sono stati alcuni mezzi utilizzati più di frequente rispetto ad altri. La scelta di analizzare i mezzi che operano su Torino per mettere in luce eventuali disparità di uso è dettata dal fatto che il territorio è meno vasto e quindi le eventuali disuguaglianze d'uso non possono essere dovute unicamente alla collocazione territoriale del mezzo.

Per poter effettuare un'analisi di questo tipo abbiamo raccolto dal database i dati inerenti ai mezzi di soccorso di Torino<sup>5</sup> ed abbiamo calcolato il numero di interventi medi realizzati da ogni singola ambulanza durante l'anno. Dopo la raccolta dei dati abbiamo calcolato una media giornaliera di utilizzo tramite una semplice formula:

$$\frac{\text{n°interventi ambulanza}}{365}$$

I dati sono i seguenti:

---

<sup>5</sup>L'informazione riguardante le ambulanze operanti sul territorio della città di Torino ci è stata fornita dai dirigenti del 118

	n° interventi	Media interventi giornalieri
MSAB h24 055	3721	10,19452
MSB Est 360	1936	5,30411
MSA h24 365	3208	8,789041
MSB Est 370	300	0,821918
MSB h24 380	5325	14,58904
MSB h24 390	6716	18,4
MSA h24 395	3297	9,032877
MSB h24 400	7259	19,88767
MSA h24 405	3031	8,30411
MSB h24 410	7716	21,13973
MSB h24 420	7153	19,59726
MSA h24 425	3794	10,39452
MSB h24 430	8663	23,73425
MSA h24 435	3726	10,20822
MSB h24 450	6985	19,13699

**Tabella 2 - Interventi delle ambulanze operanti su Tornino nell'anno 2002**

Come risulta abbastanza chiaramente dai dati della *tabella 2* prima ancora che dal grafico, vi sono alcune ambulanze che vengono usate molto meno di altre.

Tolti i casi estremi dei mezzi 360 e 370, che operano con convenzione estemporanea e quindi la differenza nel loro utilizzo potrebbe essere dovuta alle fasce orarie in cui si rendono disponibili, la differente frequenza di utilizzo degli altri mezzi può essere solo parzialmente spiegata dalla dislocazione geografica dei mezzi.

Dopo aver calcolato la media di interventi giornalieri per i mezzi che operano su Torino possiamo procedere alla costruzione del grafico a torta nel quale saranno inserite anche le percentuali di utilizzo di ogni mezzo. Tali percentuali non fanno che rafforzare l'impressione di quanto già detto sopra, ossia che l'utilizzo dei mezzi di soccorso sia piuttosto disomogeneo.

Percentuale di interventi al giorno

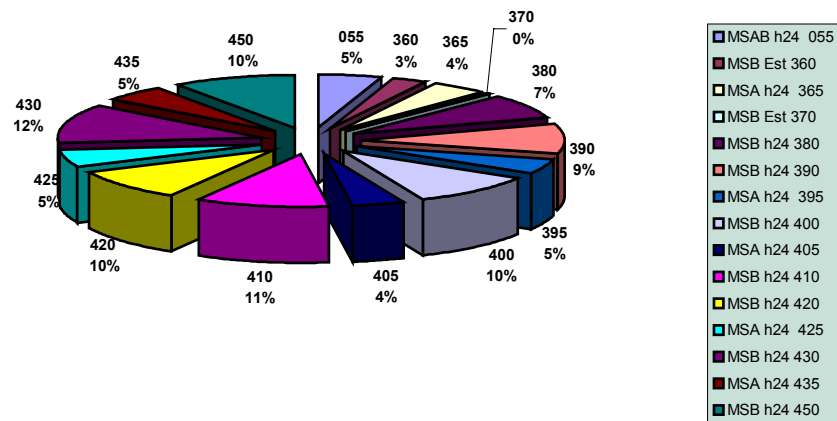


Figura 3 - Grafico di utilizzo dei mezzi di soccorso su Torino

Dopo aver sommariamente visionato i grafici ed aver evidenziato questo utilizzo disomogeneo dei mezzi di soccorso abbiamo discusso con il professore quale avrebbe potuto essere il miglior impiego dei dati in nostro possesso. In seguito alla discussione siamo giunti alla conclusione che sarebbe stato meglio ricavare dal database un giorno “*normale*”, intendendo per normale un giorno in cui non vi siano stati picchi di chiamate verso l’alto o verso il basso.

Per giungere a questo risultato abbiamo estrapolato dal database tutti gli interventi dell’anno suddividendoli in mesi e giorni e raggruppando i dati in un foglio excel strutturato come sotto:

	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato	Domenica
n interventi in gennaio	15626	574	533	480	502	479	504
	508	427	489	513	475	521	446
	457	472	461	547	528	589	521
	517	493	510	492	554	566	465
	516	487	494	506			

<b>n interventi in febbraio 14307</b>					496	569	541
	522	442	496	565	504	558	564
	562	492	484	494	477	454	546
	539	567	496	472	513	495	512
	534	441	463	509			
<b>n interventi in marzo 15540</b>					524	564	577
	520	471	470	525	507	492	548
	524	503	503	420	472	552	563
	466	463	463	533	516	549	503
	477	469	448	474	459	456	529
<b>n interventi in aprile 14665</b>	501	486	461	430	468	468	565
	512	511	471	475	498	479	506
	530	458	477	487	474	533	508
	481	448	484	518	482	537	441
	482	494					
<b>n interventi in maggio 15450</b>			461	506	471	474	494
	456	482	474	429	439	456	490
	488	468	502	497	500	466	598
	552	479	463	429	552	572	558
	496	501	541	539	617		
<b>n interventi in giugno 16400</b>						542	627
	467	517	435	463	500	491	464
	523	528	517	484	618	536	657
	590	609	685	617	559	598	655
	607	544	519	565	492	507	484
<b>n interventi in luglio 15511</b>	510	475	463	539	497	512	532
	501	525	497	516	497	522	487
	547	487	509	454	422	551	509
	555	478	510	486	502	508	526
	516	466	412				
<b>n interventi in agosto 13132</b>				467	498	514	475
	422	442	463	482	416	367	374
	412	379	414	418	444	385	426
	413	377	397	356	360	422	418
	452	420	450	430	413	417	
<b>n interventi in settembre 14825</b>							518
	467	447	425	472	536	497	526
	449	471	460	424	451	517	548
	460	511	448	476	544	536	556
	522	424	491	556	503	556	522
	512						

<b>n interventi in ottobre 15733</b>		446	528	475	431	554	556
	540	403	507	526	530	583	550
	520	471	481	472	526	560	521
	457	460	497	501	492	500	576
	545	517	514	494			
<b>n interventi in novembre 14717</b>					510	524	439
	500	477	437	451	525	497	527
	478	481	491	520	531	541	478
	504	441	430	518	494	530	478
	531	514	451	462	439	518	
<b>n interventi in dicembre 16262</b>							481
	492	467	517	477	508	501	542
	473	506	451	424	607	571	576
	546	547	550	520	555	527	533
	620	513	542	533	575	505	573
	532	498					

**Tabella 3- Suddivisione degli interventi nell'anno 2002**

Dopo aver suddiviso in questo modo il numero di interventi effettuati durante l'anno abbiamo calcolato quante chiamate sono state gestite dalla centrale operativa durante ogni settimana. Questa operazione è stata effettuata tramite una somma per riga, ossia le risposte di ogni singola giornata della centrale operativa sono state sommate in modo da ottenere gli interventi effettuati durante ogni singola settimana.

I risultati ottenuti sono riassunti in un'altra tabella, dalla quale abbiamo anche ricavato la media di interventi giornalieri nelle settimane in questione.

Il calcolo è stato effettuato in modo molto semplice:

$$\frac{\text{n°interventi settimanali}}{\text{n° effettivi giorni di lavoro durante la settimana}}$$

La motivazione che risiede nell'uso dei giorni effettivi lavorati come denominatore è molto semplice, infatti, avendo constatato che spesso la

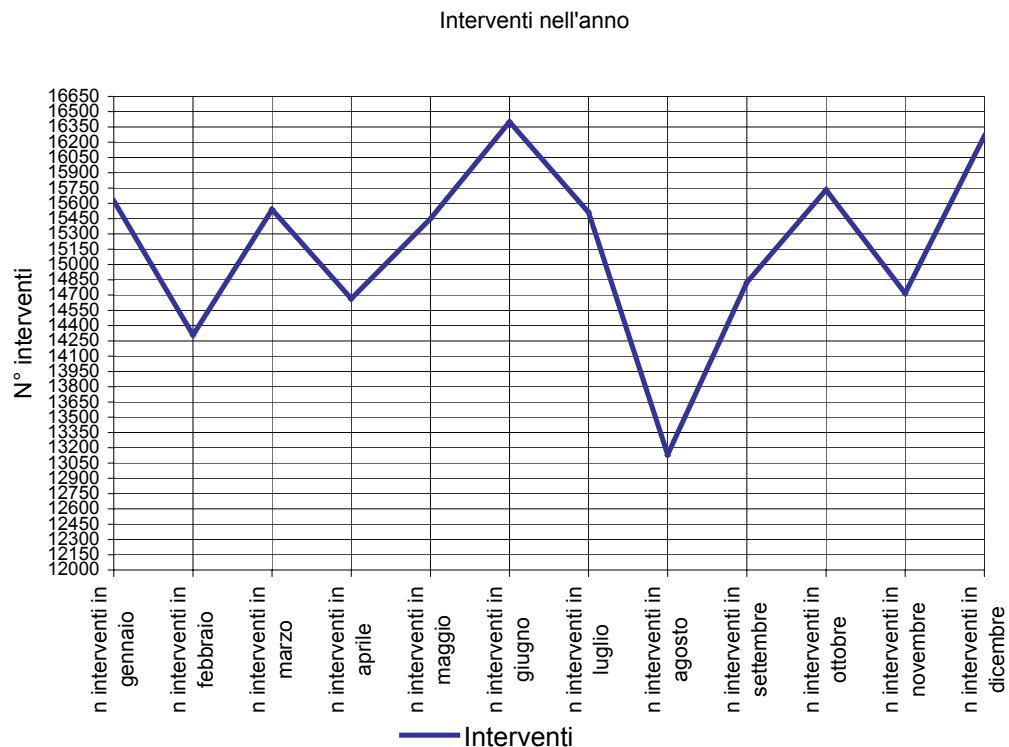
prima e l'ultima settimana del mese risultavano interrotte nel mezzo abbiamo preferito utilizzare il numero di giorni effettivi lavorati perché volevamo che l'interruzione delle settimane non incidesse in alcun modo sul numero medio di interventi giornalieri. In questo modo siamo infatti riusciti a conservare la suddivisione dei mesi.

		N° eff. gg di lavoro	N°int.sett.	Media
<b>n interventi in gennaio 15626</b>	1° settimana	6	3072	512,00
	2° settimana	7	3379	482,71
	3° settimana	7	3575	510,71
	4° settimana	7	3597	513,86
	5° settimana	4	2003	500,75
<b>n interventi in febbraio 14307</b>	1° settimana	3	1606	535,33
	2° settimana	7	3651	521,57
	3° settimana	7	3509	501,29
	4° settimana	7	3594	513,43
	5° settimana	4	1947	486,75
<b>n interventi in marzo 15540</b>	1° settimana	3	1665	555,00
	2° settimana	7	3533	504,71
	3° settimana	7	3537	505,29
	4° settimana	7	3493	499,00
	5° settimana	7	3312	473,14
<b>n interventi in aprile 14665</b>	1° settimana	7	3379	482,71
	2° settimana	7	3452	493,14
	3° settimana	7	3467	495,29
	4° settimana	7	3391	484,43
	5° settimana	2	976	488,00
<b>interventi in maggio 15450</b>	1° settimana	5	2406	481,20
	2° settimana	7	3226	460,86
	3° settimana	7	3519	502,71
	4° settimana	7	3605	515,00
	5° settimana	5	2694	538,80
<b>n interventi in giugno 16400</b>	1° settimana	2	1169	584,50
	2° settimana	7	3337	476,71
	3° settimana	7	3863	551,86
	4° settimana	7	4313	616,14
	5° settimana	7	3718	531,14

<b>n interventi in luglio 15511</b>	1° settimana	7	3528	504,00
	2° settimana	7	3545	506,43
	3° settimana	7	3479	497,00
	4° settimana	7	3565	509,29
	5° settimana	3	1394	464,67
<b>n interventi in agosto 13132</b>	1° settimana	4	1954	488,50
	2° settimana	7	2966	423,71
	3° settimana	7	2878	411,14
	4° settimana	7	2743	391,86
	5° settimana	6	2582	430,33
<b>n interventi in settembre 14825</b>	1° settimana	1	518	518,00
	2° settimana	7	3370	481,43
	3° settimana	7	3320	474,29
	4° settimana	7	3531	504,43
	5° settimana	7	3574	510,57
	6° settimana	1	512	512,00
<b>n interventi in ottobre 15733</b>	1° settimana	6	2990	498,33
	2° settimana	7	3639	519,86
	3° settimana	7	3551	507,29
	4° settimana	7	3483	497,57
	5° settimana	4	2070	517,50
<b>n interventi in novembre 14717</b>	1° settimana	3	1473	491,00
	2° settimana	7	3414	487,71
	3° settimana	7	3520	502,86
	4° settimana	7	3395	485,00
	5° settimana	6	2915	485,83
<b>n interventi in dicembre 16262</b>	1° settimana	1	481	481,00
	2° settimana	7	3504	500,57
	3° settimana	7	3608	515,43
	4° settimana	7	3778	539,71
	5° settimana	7	3861	551,57
	6° settimana	2	1030	515,00

Tabella 4 - Media interventi settimanali nell'anno 2002

Graficamente i risultati sono:



**Figura 4- Trend degli interventi durante l'anno 2002**

Dalla *Figura 4* è evidente come all'interno di un anno vi siano periodi che richiedono un impegno maggiore della centrale operativa, si veda ad esempio il periodo che va da Aprile a Giugno, ed un periodo relativamente calmo, si veda il picco verso il basso del mese di Luglio.

Per chiarire meglio gli andamenti settimanali degli interventi abbiamo creato, dagli stessi dati un grafico che ci permettesse di mettere in luce il numero di interventi di ogni singola settimana dei mesi dell'anno. In questo modo siamo riusciti a visualizzare in modo facile e diretto se all'interno dell'anno vi siano giorni della settimana maggiormente critici rispetto ad alti.

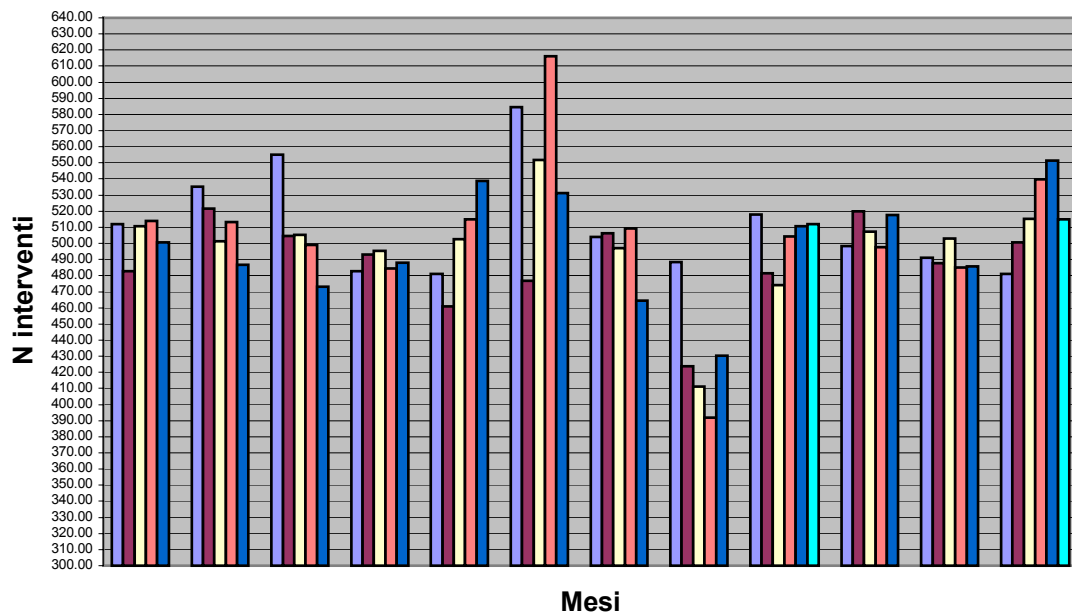


Figura 5 - Grafico interventi settimanali anno 2002

Per scegliere il giorno da noi denominato *normale* siamo andati a cercare quanto gli interventi effettuati durante ogni singolo giorno si discostassero dalla media generale dell'anno. La media generale durante l'anno è stata calcolata in modo intuitivo:

$$\frac{\text{Tot. Int.}}{\text{gg. dell'anno}}$$

e dopo un'attenta analisi siamo giunti ad isolare un giorno che è il 23 ottobre 2002, il quale risulta essere un giorno leggermente al di sotto della media .

Dopo aver stabilito il giorno dal quale saremmo partiti abbiamo iniziato ad elaborare i dati in modo che potessero essere utili ai fini rappresentativi dei fatti realmente accaduti del nostro modello.

### 7.2.3 Il trattamento dei dati incompleti

Dopo aver analizzato il database fornitoci dalla centrale operativa di Grugliasco, e dopo aver stabilito il giorno che avremmo dovuto ricostruire per testare l'attinenza del modello con la realtà abbiamo iniziato la pulizia dei dati per poter creare le ricette degli eventi.

Non appena abbiamo iniziato il lavoro di pulizia dei dati ci siamo resi conto che questi, nonostante sembrassero completi, erano in realtà ampiamente incompleti.

Dato questo stato di cose ci siamo trovati a dover pensare come poter rendere utilizzabili il maggior numero di dati possibili in modo da non approssimare troppo la realtà e quindi in modo da poter ricreare il più fedelmente possibile la giornata che avevamo preso in considerazione.

Dopo un'attenta riflessione abbiamo pensato che, per i nostri scopi, ci sarebbero potute essere utili le cartine presenti tra il materiale che ci è stato fornito dalla centrale operativa del 118. Purtroppo, e prevedibilmente, l'utilizzo delle cartine richiedeva un'elaborazione delle stesse che potesse rendere le scale della cartina di Torino e della cartina della provincia compatibili. Naturalmente un'elaborazione di questo tipo avrebbe richiesto molto tempo e avrebbe potuto creare diverse problematiche, così come inizio per l'elaborazione dei dati abbiamo pensato di utilizzare una semplice media tra i dati mancanti.

Prima di procedere alla spiegazione del trattamento dei dati incompleti è però necessario spiegare cosa si intende per dati incompleti.

All'interno del database che ci è stato fornito dal 118 e che è stato ampiamente spiegato nei paragrafi precedenti compaiono dei campi che inizialmente non ci serviranno mentre compaiono altri campi che sono necessari per una prima elaborazione del modello. Purtroppo i campi necessari alla prima lavorazione del modello sono anche i campi maggiormente affetti da imprecisioni e mancanze. I campi in questione riguardano gli orari.

Le maggiori difficoltà riguardanti gli orari sono che:

- all'interno del database nessuna voce ci dice quando realmente una chiamata giunge alla centrale operativa ma solo quando viene aperta una scheda al PVS. Quindi la prima approssimazione necessaria è il tempo di impiego dei POF;
- all'interno del database molte voci di orario, che dovrebbero derivare dalle comunicazioni continue tra ambulanze e Box Ambulanze, sono uguali. Ciò indica l'assenza di comunicazione tra l'ambulanza e la centrale operativa.

Per rendere più chiaro quanto detto fino ad ora farò un esempio prendendo all'interno del database reale un caso completo<sup>6</sup> ed un caso che presenta la seconda problematica<sup>7</sup>.

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
142413	00:26	00:29	00:32	00:46	00:54	01:14	01:24

**Tabella 5- Caso di dati completi**

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	00:05	00:07	00:39	00:39	00:39	01:49	01:49

**Tabella 6- Caso di dati incompleti**

Come è evidente dall'esempio del caso incompleto, *tabella 6*, mancano alcuni orari perché, in questo caso, l'ambulanza non ha comunicato con la centrale operativa i dati riguardanti l'orario di partenza dell'ambulanza verso il luogo dell'incidente e l'orario di arrivo sul luogo dell'incidente. Ciò che fa l'operatore del Box Ambulanze quando riceve la prima comunicazione, ed in questo caso si tratta dell'orario di partenza

<sup>6</sup> Nell'esempio riporterò solo le voci che sono utili allo scopo di evidenziare i problemi sopra elencati.

<sup>7</sup> Infatti la prima problematica si presenta sistematicamente per ogni telefonata giunta alla centrale operativa.

verso l'ospedale, è inserire l'orario comunicatogli nel campo inerente ed automaticamente gli altri campi si riempiono dello stesso orario. La stessa situazione si verifica successivamente quando l'ambulanza comunica unicamente l'orario di operatività<sup>8</sup>. Quindi i dati in realtà sarebbero così strutturati:

<b>IDScheda</b>	<b>OraScheda</b>	<b>OraAllarme</b>	<b>OraPartenza</b>	<b>OraArrivo</b>	<b>OraPartenzaH</b>	<b>OraArrivoH</b>	<b>OraOperativo</b>
000001	00:05	00:07			00:39		01:49

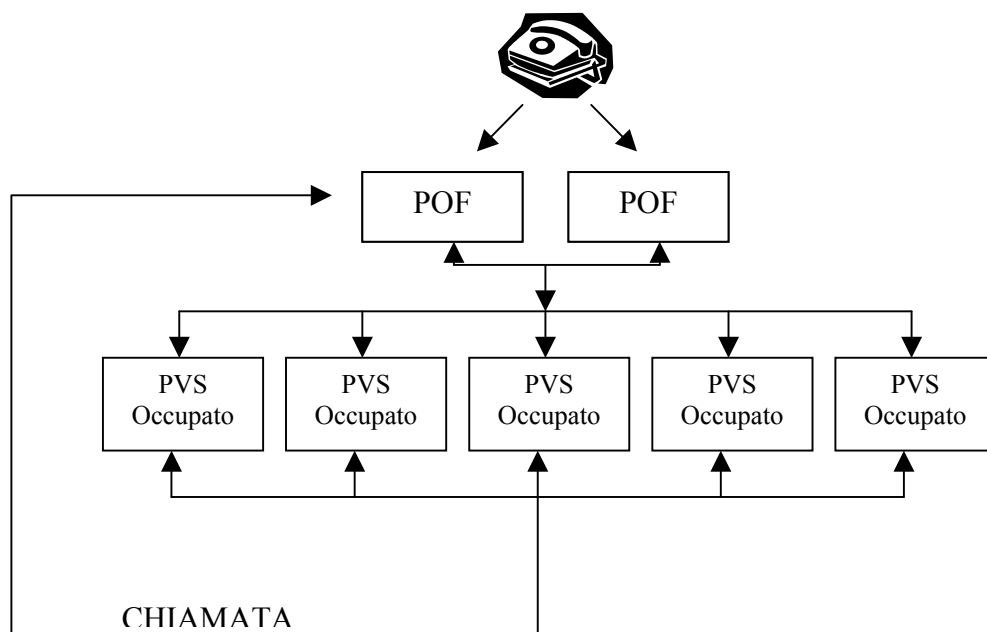
**Tabella 7 - Caso dati incompleti**

Un caso completo vedrà tutti i campi compilati con orari differenti.

Per quanto riguarda l'orario della scheda, come già detto, indica unicamente quando la chiamata arriva all'infermiere che lavora al Posto di Valutazione Sanitaria e quindi si ignorano i passaggi precedenti che possono essere brevi, nel caso l'utente trovi immediatamente un operatore libero, oppure più lunghi nel caso in cui non vi siano operatori liberi e quindi gli operatori filtro si vedano costretti ad inoltrare la chiamata manualmente. Volendo schematizzare questa situazione limite:

---

<sup>8</sup> Quando un'ambulanza si dà operativa vuol dire che da quel momento in poi la stessa potrà essere richiamata per altri interventi.



**Figura 6 - Iter della chiamata nel caso le linee PVS siano tutte occupate**

Come si può vedere dall'immagine, nel caso in cui tutti gli operatori sono occupati la chiamata torna indietro e sarà poi inoltrata manualmente dall'operatore che lavora sulla postazione filtro. È ovvio che mancando il dettaglio riguardante il POF si perdono molte informazioni nel modello.

Per ovviare a questo inconveniente nel modello stimeremo un tempo di risposta del POF ignorando i casi degli inoltri manuali che sono, comunque, eccezioni.

Per risolvere invece il problema riguardante i casi incompleti abbiamo agito in modo molto semplice, infatti lavorando sui dati ci siamo resi conto che erano sempre presenti le indicazioni riguardanti l'orario di apertura della scheda e l'orario di allarme dell'ambulanza oltre che l'orario di operatività della stessa ambulanza. Stando così le cose e prendendo per buono quanto ci era stato riferito dal direttore della centrale operativa, e cioè che all'ambulanza poteva essere affidata una nuova missione solo quando questa si rendeva disponibile<sup>9</sup>, abbiamo ritenuto che fosse una buona

<sup>9</sup> Cioè operativa

approssimazione fare una media interna tra i dati disponibili per poterne ricavare l'orario mancante. Per chiarire meglio quanto detto è necessario riportare un esempio in cui sono riportati tutti i passaggi utili per ottenere quanto ci siamo proposti.

Dato un caso incompleto, riporterò quello che ho precedentemente portato ad esempio:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	00:05	00:07			00:39		01:49

**Tabella 8 - Caso incompleto**

I primi dati mancanti sono quelli riguardanti *l'orario di partenza* dell'ambulanza verso il luogo dell'incidente e *l'orario di arrivo* della ambulanza sul luogo dell'incidente. Per ricavare questi orari abbiamo agito in questo modo:

$$\frac{(OraPartenzaH - OraAllarme)}{3} = \frac{(00:39 - 00:07)}{3} = 10.\bar{6}$$

Dato il risultato aggiungerò dieci ed undici minuti rispettivamente all' *ora allarme*, che mi è stata fornita nel database, ed all' *ora partenza*, che ho ricavato. Il risultato è:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	00:05	00:07	00:17	00:28	00:39		01:49

**Tabella 9 – Completamento del caso incompleto**

In questo caso particolare la divisione non è esatta e quindi si possono riscontrare alcuni problemi di arrotondamento che però non causano alcuna perdita di informazione.

Il passo successivo è quello di ricavare allo stesso modo il successivo orario mancante, in questo caso l'operazione è:

$$\frac{(OraOperativo - OraPartenzaH)}{2} = \frac{(01:49 - 00:39)}{2} = 35$$

Ottenendo:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	00:05	00:07	00:17	00:28	00:39	01:14	01:49

**Tabella 10 - Caso completo ricavato**

Naturalmente l'operazione è stata svolta per tutti i dati della giornata che abbiamo analizzato, in questo modo siamo riusciti ad ottenere dei casi completi.

### 7.3 La scelta delle unità

Dopo aver eseguito l'intera analisi dei dati inerenti alla giornata del 23 ottobre 2002, che è la giornata *normale* da noi scelta, abbiamo incominciato a dare una struttura al nostro modello costruendo le *Unità* del modello che sono la struttura portante dello stesso. Nel nostro caso le unità produttive non *produrranno* nulla se non servizi e quindi non ci sarà bisogno dell'utilizzo del magazzino che può a queste essere associato.

Ora analizziamo come queste unità sono state costruite.

Le unità presenti all'interno della simulazione sono state scelte tenendo conto della possibilità di generalizzare successivamente il modello tramite una simulazione della realtà del servizio 118 ex ante. Si aveva però la necessità che questa simulazione rispecchiasse per il momento la realtà,

nel senso che questa deve ricreare ciò che all'interno del mondo è accaduto. Questo approccio l'abbiamo chiamato ex-post.

Per poter capire di quali unità avevamo necessità per simulare la giornata da noi scelta abbiamo prima di tutto analizzato il funzionamento della centrale operativa individuando quali unità operanti all'interno della Centrale Operativa dovevano essere inserite nella simulazione per poter rappresentare in modo preciso il funzionamento della Centrale Operativa.

Per fare ciò abbiamo prima di tutto chiarito a noi stessi, e con l'aiuto dei dirigenti del 118, il trattamento di una chiamata.

Quello che è venuto fuori è che la chiamata segue un preciso iter:

- arriva al POF e passa da uno dei due operatori assegnati a questa postazione. Dopo aver ascoltato la richiesta del chiamante l'operatore assegnato a questa postazione inoltra la chiamata;
- la chiamata giunge al PVS e passa da una delle sei postazioni libere per valutarne la gravità. Mentre la chiamata può continuare con l'infermiere addetto alla postazione la richiesta di intervento viene inoltrata ad un'altra postazione;
- la richiesta, e non la chiamata, passa a questo punto ad uno dei due Box Emergenze. I Box si dividono le chiamate in base al luogo di provenienza, infatti uno dei due è assegnato alle emergenze sulla città di Torino mentre l'altro è assegnato alla gestione delle emergenze sulla provincia di Torino. I Box comunicano con le ambulanze e gestiscono il loro intervento;
- Le ambulanze (e nel nostro caso ogni singola ambulanza) sono singole unità che intervengono per portare a termine la missione. Queste unità chiamano poi in causa, oltre al Box altre due unità che sono l'ospedale e la postazione di chiusura;

- l'ospedale, che nel nostro caso ha tempo di lavorazione zero, è stato introdotto in prospettiva di un futuro ampliamento della simulazione;
- la postazione di chiusura è invece presente all'interno della centrale operativa ed interviene per raccogliere i dati del trasportato e per archiviare il caso;

Una volta chiarito il funzionamento della Centrale Operativa abbiamo cercato di assegnare dei “*nomi*” alle unità presenti in centrale operativa ed alle ambulanze avendo cura di semplificare il lavoro per una futura generalizzazione della simulazione.

Anzitutto abbiamo deciso che il POF era rappresentato da due unità che però rispondono alla stessa fase. Questa scelta è dettata dal fatto che dai dati non era possibile capire quale dei due avesse realmente lavorato. Questo stato di cose è presente per ogni singola chiamata arrivata alla centrale e quindi abbiamo avuto la necessità di discostarci dalla rigida rappresentazione ex post e lasciare alla simulazione un po' di libertà nella scelta dell'unità che poteva svolgere la fase di cui stiamo discutendo. Chiaramente abbiamo deciso di dare alle unità un *unitCriterion*, e cioè il modo in cui la simulazione sceglie l'unità da utilizzare, di tipo 2. Ciò vuol dire che la simulazione sceglie l'unità con la coda più corta, esattamente come succede nella realtà. Le unità sono

103 ➔ Unità POF

104 ➔ Unità POF

rispondono alla fase

100 ➔ Fase per i due POF

Lo stesso ragionamento è stato seguito per i PVS. Infatti anche per questi era impossibile risalire a chi avesse lavorato realmente e così abbiamo seguito lo stesso criterio che abbiamo adottato per i POF. Purtroppo facendo ciò abbiamo semplificato molto la realtà perché abbiamo nascosto un evento che all'interno di una centrale operativa può verificarsi. Questo evento consiste nel fatto che le chiamate con il codice rosso una volta inoltrate dal POF al PVS se non trovano nessuna linea libera ritornano al POF per poi essere inoltrate manualmente dall'operatore della postazione filtro al posto di valutazione sanitaria. Dai dati ciò non era visibile e così abbiamo deciso di non valutare il problema e lasciare che queste chiamate vadano in coda come tutte le altre se non risultano postazioni libere.

Le postazioni di valutazione sanitaria sono quindi:

- 201 ➔ Unità PVS
- 202 ➔ Unità PVS
- 203 ➔ Unità PVS
- 204 ➔ Unità PVS
- 205 ➔ Unità PVS
- 206 ➔ Unità PVS

e rispondono alla fase:

- 200 ➔ Fase per PVS

Per ciò che riguarda i Box ambulanze invece non si è posto alcun tipo di problema perché nel database che ci è stato fornito dalla centrale operativa di Grugliasco era indicato il luogo da cui proveniva la chiamata e quindi era facile dedurre quale box avesse gestito l'intervento coordinando l'azione del mezzo di soccorso e scegliendo quello più adatto all'intervento in base al tipo di mezzo necessario ed alla vicinanza dello stesso. Questo stato di cose ha fatto nascere le due unità del Box ambulanza che

rispondono a due fasi differenti. Il fatto che ogni box risponde ad una fase differente ha fatto nascere l'esigenza di non confondere troppo la simulazione. Ciò è stato ottenuto legando la fase al numero di unità, e cioè dando alla fase lo stesso numero dato all'unità che esegue la fase stessa. Nascono così le due unità che sono:

302 ➔ Unità BoxAmbulanze Torino

303 ➔ Unità BoxAmbulanze Provincia

e sono legate alla fase produttiva

302 ➔ Fase produttiva per l'unità BoxAmbulanze Torino

303 ➔ Fase produttiva per l'unità BoxAmbulanze Provincia

Subito dopo i Box nasce l'esigenza di creare le unità rappresentanti le ambulanze. In un primo momento si era pensato di dare alle unità ambulanze solo tre numeri rappresentanti i tre tipi di ambulanze a disposizione della centrale operativa, ossia le MSA, le MSAB e le MSB. Immediatamente abbiamo capito che questa non era la strada migliore perché toglieva al modello significatività e soprattutto non permetteva di fare una simulazione ex post, la quale ci dice esattamente come sono andate le cose e ci indica quali unità sono state utilizzate per farle.

La strada che abbiamo intrapreso successivamente è quella di chiamare le unità ambulanze con il numero identificativo del mezzo utilizzato dalla stessa centrale operativa. In questa fase del lavoro abbiamo legato le ambulanze ad una fase di produzione che abbiamo indicato con lo stesso numero utilizzato per indicare l'ambulanza.

Un esempio di come risultavano le unità e le fasi di produzione è:

390 ➔ ambulanza 390

legata alla fase di produzione:

390➔ fase di produzione legata all'ambulanza 390

Solo in un secondo momento si è pensato di legare la fase produttiva dell'ambulanza al tipo di mezzo aggiungendo in testa al numero della fase un numero che indicasse il tipo di mezzo. I numeri identificativi sono:

1➔ utilizzato per indicare un'ambulanza di tipo MSB

2➔ utilizzato per indicare un'ambulanza di tipo MSAB

3➔ utilizzato per indicare un'ambulanza di tipo MSA

Quindi riprendendo l'esempio di cui sopra avremo che la nostra unità produttiva rimarrà la medesima ma la fase di produzione diventa:

1390➔ fase di produzione legata all'ambulanza 390

Questo tipo di indicazione ci dice che l'ambulanza 390 è di tipo MSB.

Questa distinzione è nata dall'esigenza di rendere il modello facilmente convertibile da un modello con impostazione ex post ad un modello di tipo ex ante. È chiaro che vi sarà ancora bisogno di raffinare questo tipo di dicitura per le ambulanze perché il modello sia veramente trasportabile ex ante senza problemi. Questo tipo di trasformazione è legata al fatto che le ambulanze vanno divise in mezzi che operano in città e mezzi che operano in provincia. Probabilmente ciò verrà fatto nello stesso modo con cui sono stati divisi i mezzi in base alla loro tipologia, e cioè aggiungendo un numero in testa che sia indice della localizzazione dell'ambulanza.

Per ora è attivo solo il controllo sul tipo di mezzo e nella simulazione le unità ambulanza e le loro fasi sono:

Amb.	Fase	Amb.	Fase	Amb.	Fase	Amb.	Fase
1	3001	390	1390	300	1300	770	1770
55	2055	395	3395	301	1301	775	2775
65	2065	400	1400	320	1320	780	1780
90	1090	405	3405	330	1330	800	1800
100	1100	410	1410	335	2335	850	1850
110	1110	420	1420	345	3345	905	3905
125	3125	425	3425	350	1350		
135	3135	430	1430	360	1360		
160	1160	435	3435	361	1361		
210	1210	450	1450	365	3365		
230	1230	510	1510	380	1380		
240	1240	520	1520	630	1630		
245	2245	555	3555	655	3655		
260	1260	580	1580	680	1680		
285	3285	610	1610	685	3685		
290	1290	620	1620	750	1750		

Tabella 11- Unità ambulanze e fasi operative

Infine sono state create le ultime due unità che però non creano alcun tipo di problema in quanto sono entrambe unità singole. Le due unità sono l'unità Ospedale e l'unità Postazione di chiusura. Queste sono indicate così:

701 ➔ Unità Ospedale

801 ➔ unità Chiusura Servizio

le fasi di produzione sono:

700 ➔ Fase dell'unità Ospedale

801 ➔ Fase dell'unità Chiusura servizio

## 7.4 La scelta del tempo

In seguito alla costruzione della struttura portante del modello il passo successivo era quello di stabilire una suddivisione temporale che potesse andare bene per tutte le esigenze del modello.

Inizialmente, dati tempi molto lunghi di intervento delle ambulanze eravamo tentati di utilizzare i minuti come suddivisione temporale ma in un secondo tempo la scelta del tempo è ricaduta su una misura temporale un po' diversa, ossia i decimi di minuto. Il motivo di questa bizzarra scelta è dettato dall'esigenza di simulare anche l'operatività del POF che presumibilmente non è molto estesa. Questa ipotesi nasce dall'esperienza personale di alcuni componenti del gruppo che hanno lavorato come centralinisti e che hanno assicurato che il tempo di ricevere ed inoltrare una chiamata non può essere superiore ai 10 secondi, e quindi ad un tick della nostra simulazione.

La scelta di questa suddivisione temporale ha comportato una serie di cambiamenti molto grandi all'interno dei dati da noi corretti e stimati come già detto sopra. Infatti questi, essendo indicati in distinzioni temporali di ore e minuti, devono essere trasformati in decimi di secondi. Il metodo per fare ciò è molto semplice.

Il primo passo è stato quello di stabilire l'ordine di partenza di ogni ricetta, in modo da poter elencare l'ordine esatto delle ricette all'interno del file *orderStartingSequences.xls* che rappresenta appunto la sequenza temporale degli eventi. Per ottenere i decimi minuto esatti abbiamo moltiplicato il dato per sei, che è appunto il numero di decimi di minuto presenti in un minuto. Ora riprenderò l'esempio di prima e ripeterò l'operazione svolta per chiarire come è avvenuta la trasformazione dei dati.

Dato l'evento identificato con il numero 000001 ed opportunamente riempito nelle sue parti mancanti:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	00:05	00:07	00:17	00:28	00:39	01:14	01:49

Tabella 12 - Caso riempito delle parti mancanti

La prima trasformazione avviene sull'*ora scheda* in questo modo:

$$OraScheda \times 6 = 5 \times 6 = 30$$

Quindi andando a modificare all'interno della tabella dei dati otteniamo:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	30	00:07	00:17	00:28	00:39	01:14	01:49

Tabella 13 - Modifica dell'ora scheda

Che è il tick di partenza di questa scheda, le operazioni successive consistono nel trasformare il tempo successivo in tick della simulazione. I tick successivi hanno però un significato diverso dal primo riportato, infatti indicano quanto tempo dura la fase produttiva dell'unità.

Ora riporto la procedura necessaria alla trasformazione di un unico passo successivo alla trasformazione dell'orario della scheda, questo perché tutti gli altri passi sono fatti in modo analogo.

La procedura è:

$$(OraAllarme - OraScheda) \times 6 = (00.07 - 00.05) \times 6 = 2 \times 6 = 12$$

Quindi effettuando tutte le trasformazioni otteniamo:

IDScheda	OraScheda	OraAllarme	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	30	12	60	66	66	210	210

Tabella 14 - Trasformazione di tutti i dati in decimi di minuto

Quest'operazione verrà ripetuta per tutte le ricette e ci restituirà i dati che vanno all'interno delle stesse anche se si devono ancora apportare delle piccole modifiche di cui parlerò nel prossimo paragrafo.

## 7.4 La costruzione delle ricette

Successivamente alla modifica temporale dei dati forniteci abbiamo incominciato a costruire le ricette necessarie per permettere alla simulazione di funzionare e soprattutto necessarie ai fini della costruzione del nostro modello ex post della centrale operativa.

Lo sviluppo delle ricette ci ha portato subito ad evidenziare alcune problematiche inerenti al funzionamento della centrale operativa e difficilmente trasportabili nel modello. Le soluzioni ci hanno portato alla costruzione di due modelli differenti che sono l'uno l'evoluzione dell'altro e soprattutto il secondo è importantissimo per la trasposizione del modello da ex post ad ex ante.

In questa sede analizzeremo lo sviluppo di questi modelli cercando di illustrare nel modo più chiaro possibile il motivo che ci ha spinto a creare la soluzione successiva e a non accontentarci della prima versione del modello. Le due versioni sono:

- Ricette senza passi computazionali;
- Ricette con passi computazionali.

### 7.5.1 Le ricette senza passi computazionali

Per creare una ricetta abbiamo sfruttato i dati che ci erano stati forniti dalla centrale operativa debitamente trasformati come già detto precedentemente. La creazione della ricetta ci ha posto di fronte ad alcuni problemi di cui in questo capitolo abbiamo già discusso ma di cui, fino ad ora, non si è detto come sono stati affrontati.

Prima di affrontare qualsiasi tipo di problematica abbiamo stabilito come sistemare la parte formale della ricetta, ossia abbiamo stabilito il nome delle ricette ed abbiamo dato il numero identificativo della ricetta. Il nome attribuito alle ricette è *ex-post* seguito da un numero crescente. La scelta del nome *ex-post* è dovuta ad una ragione molto semplice riguardante il fatto che le ricette create per questo modello sono ricette che riproducono quanto avvenuto realmente all'interno della centrale operativa nel giorno da noi studiato.

Il numero identificativo della ricetta è invece tratto dal database del 118 ed indica il numero dato a quel evento dalla centrale operativa depurato delle prime tre cifre<sup>10</sup>. Questo numero deve poi essere inserito all'interno dei files di *sequences* per indicare la sequenza temporale dei lanci delle ricette. Volendo esemplificare quanto detto avremo:

#	Recipes	;
	ex-post1	409
	ex-post2	481
	ex-post3	488
	ex-post4	507
	...	...

**Tabella 15 - Estratto del file recipes**

<sup>10</sup> Gli eventi nella centrale operativa sono numerati con sei cifre, noi nelle nostre ricette usiamo solo le ultime tre cifre. Quindi se la numerazione della centrale dovesse essere 142333 la nostra ricetta avrebbe numerazione 333.

Come si vede dalla *tabella 15* abbiamo il nome della ricetta ed il numero identificativo della stessa.

Dopo aver risolto le parti formali della costruzione delle ricette ci siamo trovati ad affrontare i problemi reali di rappresentazione del modello.

Il primo problema che ci si poneva di fronte riguardava il tempo di operatività del POF, di cui non avevamo informazioni come già ampiamente detto. Per affrontare questa problematica abbiamo detto che al POF sarebbe stato attribuito un tempo standard di dieci secondi e quindi di un tick<sup>11</sup> di simulazione. Per non modificare in alcun modo la simulazione il tempo andava tolto da qualche parte e a noi la soluzione migliore è sembrata quella di far partire la simulazione un tick prima di quanto in realtà è segnato nei dati<sup>12</sup> forniteci. Questa approssimazione ci è sembrata la più realistica dato che realmente l'orario di arrivo della chiamata è antecedente l'orario segnato sulla scheda riguardante l'evento.

Volendo esemplificare e ritornando all'esempio standard abbiamo:

IDScheda	Inizio	PVS	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	30	12	60	66	66	210	210

**Tabella 16 - Caso con il tempo trasformato in decimi di secondo**

In questo caso la ricetta che avrebbe dovuto partire al tick 30, ossia alle ore 00:05, partirà invece al tick 29. La ricetta incomincia a prendere forma ed è:

ex-post1	001	100	s	1
----------	-----	-----	---	---

**Tabella 17 - Primo step della ricetta**

<sup>11</sup> Ricordo che un tick di simulazione corrisponde a dieci secondi.

<sup>12</sup> Ricordo che i dati non riportano il tempo di operatività del POF perché l'orario segnato all'interno dei dati riguarda unicamente l'orario indicante l'arrivo della chiamata al PVS.

In questa ricetta vediamo il nome attribuitogli, *ex-post1*, ed il numero che la identifica, *001*. In questa semplice ricetta vediamo inoltre che l'unica fase ad essere svolta è la fase 100, e cioè quella che indica l'operatività del POF, e che questa fase lavora per un tick.

Nell'*orderStartingSequences.xls* avremo:

	...			
27	;			
28	;			
29	001 *	1;		
30	;			
	...			

**Tabella 18 - Estratto dall'*orderStartingSequences.xls***

Anziché:

	...			
27	;			
28	;			
29	;			
30	001 *	1;		
	...			

**Tabella 19 - Estratto dall'*orderStartingSequences.xls* nel caso non venga attribuito al POF un tick di operatività**

Successivamente verrà riportata l'operatività del PVS. A questo punto il problema che si pone davanti è quello delle comunicazioni che esistono tra Box Ambulanze ed ambulanze, infatti anche questo dato non è riportato all'interno dei dati che ci sono stati rilasciati dalla centrale operativa. Per risolvere questa problematica la soluzione più semplice ci è sembrata, anche in questo caso, quella di togliere un tick dalla fase successiva per inserire la comunicazione all'interno della ricetta. Quindi la nostra ricetta è ora divenuta:

ex-post1	001	100	s	1	200	s	12	302	s	1	3115	s	59	302	s	1	3115	s	65	
		302	s	1	3115	s	65	302	s	1	3115	s	209	302	s	1	3115	s	209	;

**Tabella 20 - Ricetta completa senza operazione di chiusura**

Dato il solito esempio:

IDScheda	Inizio	PVS	OraPartenza	OraArrivo	OraPartenzaH	OraArrivoH	OraOperativo
000001	30	12	60	66	66	210	210

**Tabella 21 - Caso completato**

Come si può vedere dal confronto tra quello che è la ricetta, *tabella 20*, e quello che è il caso da noi analizzato, *tabella 21*, sono stati eliminati dei tick nelle varie fasi per simulare la comunicazione.

Dopo aver risolto questa problematica resta ancora da risolvere la problematica inerente all'operazione di chiusura, che, come negli altri casi, non è reperibile in nessun dato presente all'interno del database e quindi ci siamo trovati ad agire nello stesso identico modo che ci ha permesso di completare i dati mancanti del database. In questo caso, essendo l'operazione di chiusura più lunga, abbiamo tolto sei tick dall'ultima fase dell'ambulanza in modo da poter rendere meglio questo aspetto. La ricetta diviene:

ex-post1	001		100	s	1		200	s	12		302	s	1		3115	s	59		302	s	1		3115	s	65
			302	s	1		3115	s	65		302	s	1		3115	s	209		800	s	0		700	s	6
			3115	s	203		302	s	1	;															

**Tabella 22 - Ricetta completa**

L'evidente problema di questo tipo di formazione delle ricette riguarda principalmente l'interazione tra Box Ambulanze ed ambulanze.

Impostando le ricette in questo modo non potevamo pensare in alcun modo di trasformare il modello ex-post, che racconta quanto avvenuto in questa giornata, in un modello ex-ante. Con questa impostazione infatti abbiamo simulato la comunicazione tra i due agenti simulando una sosta dell'agente ambulanza per comunicare. Il problema che si sarebbe potuto verificare in un'impostazione ex-ante del modello è che l'ambulanza durante questa sosta venga chiamata per un altro servizio, dato che in quel preciso istante l'ambulanza risulta essere libera, cosa che invece non può accadere e quindi deve essere evitata all'interno del modello.

La soluzione di questo problema ci è stata fornita da uno strumento di jES che prende il nome di *passi computazionali*. Vediamo come si sviluppano le ricette con i passi computazionali e come questo strumento, che come già detto è presente in jES, è stato modificato per adattarlo alle esigenze della simulazione.

### **7.5.2 Le ricette con passi computazionali**

La trasformazione delle ricette tradizionali in ricette che contengono i passi computazionali è avvenuta senza particolari traumi, infatti le operazioni compiute sono state:

- sostituire le fasi inerenti alla comunicazione tra Box ambulanza ed ambulanza con il riferimento ad un passo computazionale;
- restituire i tick che nella fase precedente erano stati tolti per poter in qualche modo simulare la comunicazione.

Quindi dopo le opportune modifiche le ricette hanno preso questa forma:

ex-post1	001		100	s	1		200	s	12	c	1002	1	0	3115	s	60	c	1002	1	2	3115	s	66
			c	1002	1	2	3115	s	66	c	1002	1	2	3115	s	210	800	s	0		700	s	6
			c	1002	1	0	3115	s	204	;													

Tabella 23 - Ricetta completa con passi computazionali

Come si vede chiaramente ora le comunicazioni tra il Box e le ambulanze sono “mascherate” dell’uso di questo strumento.

Vediamo come questo strumento è stato sviluppato ed il significato dei numeri che compaiono con l’immissione dei passi computazionali all’interno delle ricette.

### 7.5.2.1 La creazione dei passi computazionali

I passi computazionali da noi creati dovranno attivare una ricetta, indicata all’interno delle matrici che i passi computazionali vanno a richiamare, integrando nell’OrderDistiller i dati dell’OrderStartingSequence o dell’OrderSequence.

Come risultato, idealmente, si avrà l’aggiunta in coda alla riga dell’OrderStartingSequence, processata al momento, di un secondo evento, che fa riferimento alla ricetta di cui sopra.

L’OrderDistiller lancia le ricette e poi le lascia al loro destino; jES opera in modo distribuito, con ogni ricetta che procede, come gli avvenimenti nella realtà, indipendentemente; non ha controllo centralizzato; la ricetta madre è eseguita passando di unit in unit e così si esegue un passo computazionale che fa sì che OrderDistiller lanci una ricetta.

Ciò che accade quando la simulazione è lanciata è che la gestione degli ordini nelle unità incontra, durante la lettura delle ricette, il passo computazionale generando un’informazione che sta sospesa in OrderDistiller sino al momento di eseguire il prossimo lancio di una ricetta.

Ciò che si è dovuto creare è uno specifico passo che facesse un set in OrderDistiller con il codice tratto dalla matrice usata.

La costruzione di questo meccanismo ha posto dei problemi; bisognava infatti capire dove l'OrderDistiller andasse a cercare il numero relativo alla ricetta da eseguire. La soluzione più appropriata è sembrata quella di custodire quest'informazione all'interno di un vettore posizionato in OrderDistiller stesso. Questo vettore è composto esclusivamente da zeri all'inizio della simulazione; man mano che la simulazione procede, e che i passi computazionali vengono eseguiti, esso viene riempito con i numeri che corrispondono alle ricette. OrderDistiller legge all'interno del vettore e se trova i numeri delle ricette le esegue e svuota il vettore.

Nello sviluppo del formalismo si è deciso di procedere a piccoli passi in modo da testare, volta per volta, la correttezza dei cambiamenti. Durante la prima fase sono stati creati nuovi passi computazionali inseriti in ComputationalAssembler.java<sup>13</sup>. La prima versione del file svolgeva una funzione piuttosto elementare. Definiva una matrice, ne leggeva l'indirizzo di memoria e faceva un set nella matrice. Ossia, nella prima versione all'interno della nostra matrice veniva scritto il numero 287 nelle posizioni di riga e colonna (0,0). Alla fine il contenuto della matrice veniva visualizzato a video per permettere di capire se il passo computazionale fosse stato compiuto o meno.

Tutto questo è stato creato aggiungendo le seguenti righe di codice nel file:

```
public void c1001()
{
    if(pendingComputationalSpecificationSet.
```

---

<sup>13</sup> Si deve copiare ComputationalAssembler.java nella cartella principale del programma da src/; questo perché il comando make run usa le classi contenute in lib/jesframe.jar, che sono quelle contenute in src/, ma le classi in ./, cioè la cartella corrente, sovrascrivono quelle in jesframe.jar).

```
        getNumberOfMemoryMatrixesToBeUsed()!=1)
    {
        System.out.println("Code -1001 requires three matrix; " +
            pendingComputationalSpecificationSet.
            getNumberOfMemoryMatrixesToBeUsed() +
            " found in order # " +
            pendingComputationalSpecificationSet.
            getOrderNumber());
        MyExit.exit(1);
    }

    mm0=(MemoryMatrix) pendingComputationalSpecificationSet.
        getMemoryMatrixAddress(0);

    mm0.setValue(layer,0,0,287.0);
    mm0.print();

    done=true;
} // end c1001
```

Il metodo esegue una parte di controllo comune a tutti i passi computazionali<sup>14</sup> nella quale cambia unicamente il riferimento al passo computazionale.

Inizialmente si era pensato di creare tanti passi computazionali simili a quello appena descritto, con l'unica differenza che invece di visualizzare a video il numero identificativo della ricetta, lo avrebbero scritto nel vettore

---

<sup>14</sup> if(pendingComputationalSpecificationSet.  
getNumberOfMemoryMatrixesToBeUsed()!=1)  
{  
 System.out.println("Code -1001 requires three matrix; " +  
 pendingComputationalSpecificationSet.  
 getNumberOfMemoryMatrixesToBeUsed() +  
 " found in order # " +  
 pendingComputationalSpecificationSet.  
 getOrderNumber());  
 MyExit.exit(1);  
}

contenuto in OrderDistiller; in un secondo momento si è invece deciso di modificare leggermente l'impostazione del formalismo. Si è infatti preferito creare un passo computazionale<sup>15</sup> che si occupasse unicamente di inserire in più matrici i numeri delle ricette indipendenti che simulano la comunicazione. I passi computazionali interagiscono con l'OrderDistiller riempiendo volta per volta il vettore.

Per fare ciò però era necessario conoscere la struttura di OrderDistiller, in quanto è questa classe che, attraverso il metodo Distill, si occupa del lancio delle ricette. Di seguito vengono illustrati i metodi contenuti nella classe OrderDistiller.

Dopo il costruttore che è:

```
public OrderDistiller (Zone aZone, int msn, int msl, ListImpl ul,
                      ListImpl eul, ListImpl ol, int tln, ESFrameModelSwarm mo,
                      AssigningTool at)
{ //pt
    super(aZone, msn, msl, ul, eul, ol, tln, mo, at);

    unitList=ul;
    endUnitList=eul;
    orderList=ol;
    assigningTool=at; //pt
}
```

viene descritto il metodo distill<sup>16</sup>. Tramite questo vengono lanciati gli ordini contenuti nelle ricette secondo la sequenza contenuta nell'OrderStartingSequence o nell'OrderSequence, a seconda delle esigenze delle singole simulazioni. Il controllo di quale dei due utilizzare avviene subito:

```
if(firstTime == true)
```

---

<sup>15</sup> nello specifico si tratta del passo 1001

<sup>16</sup> Il codice completo di OrderDistiller è in Appendice

```
{
    currentOrderSequenceWorksheet=orderSequences1;
    firstTime = false;
}
else
    currentOrderSequenceWorksheet = orderSequences2;
```

La variabile `firstTime` indica se il file da usare è l'`orderStartingSequence` (`orderSequences1`) o l'`orderSequence` (`orderSequences2`). Il metodo effettua altri controlli e solo successivamente arriva alla lettura delle ricette:

```
//Reading the order from the worksheet
readOrderFrom(orderSequenceWorksheet);//see below
//Checking if the recipe code exists
boolean recipeCodeFound = false;
for (int r = 0; r < recipeList.getCount() && !recipeCodeFound; r++)
{
    aRecipe = (Recipe) recipeList.atOffset(r);
    if(aRecipe.getRecipeCode() == currentOrder[0])
        recipeCodeFound = true;
}
```

ed al loro lancio:

```
for(int q = 0; q < currentOrder[1]; q++)
{
    orderCount++;
    anOrder=newOrder(getZone(),orderCount,Globals.env.getCurrentTime(),
aRecipe.getLength(),aRecipe.getRecipeSteps(),          eSFrameModelSwarm,
endUnitList);

    anOrder.setRecipeName(aRecipe.getRecipeName());
// setting the layer (from 0 to totalLayerNumber-1)
if(currentLayer <0 || currentLayer >= totalLayerNumber)
{
```

```
        System.out.println("A layer in the sequence is not included\n"+"in
the interval from 0 to totalLayerNumber-1");
        MyExit.exit(1);
    }
    anOrder.setOrderLayer(currentLayer);
    // add the active orders to the general order list (they will be
    // eliminated when dropped in a unit, being finished); this
    // list has been introduced for accounting purposes [may be it would
    // be better substitute it with a get to the units to know their waiting lists]
    orderList.addLast(anOrder);
    // sending the order to the first production unit (we are acting as the Front //End of
the ES)
```

In particolare il metodo deve controllare, ogni volta che trova un “;”, se ci sono in coda passi computazionali da eseguire, nel qual caso li manda in esecuzione prima di passare alla riga della sequence successiva.

Chiaramente una volta costruiti i passi computazionali e verificato il loro funzionamento è stata creata l’interazione tra ComputationalAssembler ed orderistiller, in modo da poter veramente creare il formalismo delle ricette che lanciano ricette.

La creazione del vettore nella classe OrderDistiller (opportunamente copiata nella directory principale) consta in due passaggi:

- la dichiarazione  
`int[] computationalRecipes;`
- la creazione vera e propria  
`computationalRecipes new Int [1000];`

La dichiarazione avviene nelle proprietà generali della classe, mentre la sua creazione avviene nel costruttore di OrderDistiller; sempre nel costruttore si ha l’inizializzazione a zero di tutte le posizioni del vettore.

In seguito si è passati alla modifica di `ComputationalAssembler` in modo da poterlo fare interagire con `OrderDistiller`.

Le attuali modifiche sono state effettuate per dotare `ComputationalAssembler` dell'indirizzo di memoria di `OrderDistiller`, in modo da permettere ad un metodo contenuto nella prima classe di andare a inserire dati nel vettore `computationalRecipes` contenuto nella seconda.

Nelle proprietà generali di `ComputationalAssembler` è stata definita una variabile `myOrderDistiller` di tipo `OrderDistiller`; `OrderDistiller` è stato inserito tra i parametri della classe `ComputationalAssembler` attraverso la variabile `od`, e nel costruttore si è assegnato il valore di `od` a `myOrderDistiller`.

Lo stesso lavoro è stato fatto nella classe `ComputationalAssemblerBasic`.

Dopo aver modificato i parametri da passare alla classe `OrderDistiller` si è dovuto modificare anche la classe `ESFrameModelSwarm`; ogni volta che si incontrava le seguenti righe di codice

```
aComputationalAssembler = new ComputationalAssembler  
(getZone());
```

si è dovuto aggiungere anche `myOrderDistiller`, in questo modo:

```
aComputationalAssembler = new ComputationalAssembler  
(getZone(), myOrderDistiller);
```

Dopo queste modifiche `Swarm` non dà più problemi di compilazione. Il problema riguardante l'interazione dovrebbe quindi essere risolto; ora si procederà alla creazione del metodo `setComputationalRecipes` che si occupa della scrittura nel vettore.

L'ultima modifica fatta è stata quella di creare un file chiamato *recipesFromRecipes.txt* in cui vengono assegnati alle matrici chiamate `M0`,

M1<sup>17</sup>, numeri identificativi delle ricette che devono essere lanciate senza dover per questo andare a modificare ogni volta il codice java inerente al formalismo dei passi computazionali che è utilizzato per il lancio di ricette.

---

<sup>17</sup> I nomi generali delle matrici accettati dal simulatore sono M0, M1, M2, M3, M4, M5, M6, M7, M8, M9