



Model de simulació per coordinar serveis d'emergència

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Jordi Sumalla Botella
i dirigit per
Daniel Guimarans Serrano
Bellaterra, 29 de gener de 2008

SUMARI

SUMARI.....	1
1. INTRODUCCIÓ.....	4
1.1 ABAST DEL PROJECTE.....	4
1.2 OBJECTIUS.....	5
2. DISSENY DEL MODEL	6
2.1. INTRODUCCIÓ.....	6
2.2. XARXES DE PETRI.....	7
2.2.1. Xarxes de Petri ordinàries.....	7
2.2.2. Xarxes de Petri Acolorides.....	9
2.3. APROXIMACIÓ AL MODEL.....	12
2.3.1. Model Flow Chart.....	12
2.3.2. Central de Dades.....	13
2.4. DEFINICIÓ DELS NODES PRINCIPALS I ELS SEUS ATRIBUTS O COLORS.....	13
2.5. DISSENY DE LES XDPA PER SEPARAT	16
2.5.1. XdPA del cos de la Policia.....	16
2.5.2. XdPA del cos de Bombers.....	19
2.5.3. XdPA del cos de les Ambulàncies.....	22
2.5.4. Submodel del moviment dels vehicles.....	25
2.5.5. Actualització de les dades I: enviament de reforços i flags de control.....	26
2.6. XDPA GENERAL.....	31
2.6.1. Actualització de les dades II.....	35
2.7. SIMPLIFICACIÓ DE LA XDPA GENERAL	38
3. SIMULACIÓ EN ARENA.....	41
3.1. QUÈ ENTENEM PER SIMULACIÓ?	41
3.1.1. Etapes d'un projecte de simulació.....	41
3.1.2. Avantatges i inconvenients d'una simulació.....	42
3.1.3. Aplicacions	43
3.1.4. Llenguatges de programació.....	43
3.2 QUÈ ÉS ARENA?	44
3.3 CONSTRUCCIÓ DEL MODEL	46
3.3.1. Línies d'acció: Consideracions prèvies	46
3.3.2. Creació d'accidents.....	47
3.3.3. Creació de les flotes de vehicles.....	48
3.3.4. Entrada línies d'acció: mòdul Decide.....	48
3.3.5. Línia d'assignació	49
3.3.6. Línia de moviment dels vehicles	50
3.3.7. Línia d'actualització.....	51
3.3.8. Línia de vehicles ocupats.....	51
3.3.9. Línia d'intervenció	52
3.3.10. Línia de finalització dels vehicles.....	53
3.3.11. Línia d'actualització de les dades.....	56
3.4. REPRESENTACIÓ FINAL DEL MODEL DE SIMULACIÓ	57

4. CONNEXIÓ BDD-CORBA	59
4.1. ESPECIFICACIÓ DE LES TAULES	59
4.2. PLATAFORMES DISTRIBUÏDES	64
4.2.1. <i>Introducció</i>	64
4.2.2. <i>CORBA</i>	64
4.2.3. <i>Connexió de components via CORBA mitjançant una DLL</i>	65
4.2.4. <i>Accés a la Base de Dades i invocació a la DLL</i>	65
4.2.5. <i>Funcions disponibles</i>	66
5. CONSTRUCCIÓ DEL MODEL VBA	68
5.1. SIMILITUDS I CANVIS RESPECTE AL MODEL DE SIMULACIÓ	68
5.2 LÍNIES D'ACCIÓ I BLOCS VBA	69
5.2.1. <i>Creació d'accidents</i>	69
5.2.2. <i>Línia d'actualització</i>	71
5.2.3. <i>Línia d'assignació</i>	71
5.2.4. <i>Línia d'assignació i moviment dels vehicles</i>	72
5.2.5. <i>Línia d'intervenció</i>	73
5.2.6. <i>Línia de finalització del servei</i>	74
5.3 REPRESENTACIÓ FINAL DEL MODEL VBA	75
6. CONCLUSIONS	77
BIBLIOGRAFIA	79
ANNEXES	
ANNEX A. SISTEMES I MODELS: DESCRIPCIÓ I CLASSIFICACIÓ	81
A.1. SISTEMES	81
A.2. MODELS	82
A.2.1. <i>Classificació dels models</i>	83
ANNEX B. ESTRUCTURA DEL PROGRAMA ROCKWELL ARENA	84
B.1. DISPOSICIÓ DE LES FINESTRES	84
B.2. ESTRUCTURA DELS MÒDULS	85
B.2.1. <i>Basic Process Panel</i>	86
B.2.2. <i>Advanced Process Panel</i>	87
B.2.3. <i>Advanced Transfer Panel</i>	88
B.2.4. <i>Blocks Panel (SIMAN template)</i>	88
B.3. FUNCIONS DE DISTRIBUCIÓ DE PROBABILITATS	89
B.3.1. <i>Distribució Normal</i>	90
B.3.2. <i>Distribució Triangular</i>	91
B.3.3. <i>Distribució Uniforme</i>	91
B.3.4. <i>Distribució Exponencial</i>	92
ANNEX C. IMPLEMENTACIÓ DEL MODEL VBA	93
C.1. LENGUATGE SIMAN: BLOCS VBA	93
C.2. FITXERS KML GOOGLEEARTH	93
C.2.1. <i>Definició i estructura d'un fitxer KML</i>	93
C.2.2. <i>Generació del Moviment al Google Earth</i>	94

C.3. CODI VISUAL BASIC	96
C.3.1. <i>Generació d'Accidents</i>	96
C.3.2. <i>Actualització dels Accidents</i>	99
C.3.3. <i>Generació de les flotes dels serveis</i>	99
C.3.4. <i>Policia</i>	104
C.3.5. <i>Bombers</i>	111
C.3.6. <i>Ambulàncies</i>	117

1. INTRODUCCIÓ

Quan es produeix un accident en carretera tenim dos factors que esdevenen primordials per minimitzar el temps de resposta i optimitzar els recursos: coordinació i assignació.

La coordinació esdevé difícil i complexe donat que la informació que disposem al principi de l'accident no és completa i va variant en el transcurs d'aquest. Per una banda, necessitem una coordinació interna de cada servei i per l'altra i més complicada, una coordinació entre tots els serveis per obtenir uns resultats òptims.

Una bona assignació dels vehicles que conformen els serveis d'emergència (policia, bombers i ambulàncies) és necessària per disposar sempre dels efectius per cobrir les necessitats de l'accident.

Aquest projecte presenta una definició i implementació d'un model de simulació basat en aquests dos factors comentats.

La definició del model es realitzarà mitjançant les Xarxes de Petri Acolorides mentre que les simulacions es faran amb el programa Rockwell Arena 7.0.

La primera simulació constitueix un model teòric del problema mentre que la segona s'aproxima més a la realitat ja que introduïm informació real de flotes i vehicles a partir d'una base de dades. Aquesta connexió es realitza mitjançant la plataforma Corba.

1.1 Abast del projecte

Aquest projecte forma part d'un altre més ampli dins del departament de Telecomunicació i d'Enginyeria de Sistemes (TES). En aquest projecte [Guimarans, et al., 2006], s'estudien les millors eines de decisió per fer front al problema de la coordinació dels serveis a partir de la optimització d'algoritmes. Aquests algoritmes, combinen mètodes heurístics i tècniques satisfactòries als problemes de restricció.

Per tant, l'aspecte econòmic careix d'importància ja que l'objectiu no és la implantació en empresa. El que sí té importància del projecte presentat, és la validació d'eines susceptibles de ser implantades.

1.2 Objectius

L'objectiu principal del projecte és definir i implementar un model de simulació basat en la coordinació i assignació dels serveis d'emergència en accidents de trànsit.

Per assolir-lo, s'han marcat diferents passos. Donat que aquest és un projecte on la resolució de cada pas és necessari per poder abordar el següent, cada pas també s'ha dividit en petits objectius.

Els passos que s'han seguit són:

1. Anàlisi del problema per al posterior disseny
2. Disseny de les Xarxes de Petri Acolorides
 - * Primer disseny de cada servei per separat
 - Actuació del servei i coordinació amb l'accident
 - Actualització de les dades i enviament de reforços
 - * Submodel de moviment
 - * Disseny global dels tres serveis
 - Coordinació entre els serveis i amb l'accident
 - Actualització de les dades i enviament de reforços
3. Simulació en Arena
 - * Creació de les cues d'accidents i vehicles
 - * Línies d'acció
 - * Actualització i bloqueig
4. Base de dades- Connexió plataforma Corba
 - * Creació de les taules que conformen la Base de Dades
5. Simulació VBA en Arena
 - * Creació dels blocs VBA i implementació
 - * Línia de moviment
 - * Creació fitxers kml i simulació amb GoogleEarth

2. DISSENY DEL MODEL

2.1. Introducció

El model de la simulació es farà usant el software Rockwell Arena 7.0. Abans, però, és necessari realitzar un estudi de què i com ho volem fer. No cal dir que intentar fer una simulació de l'actuació dels serveis d'emergència en carretera sense un estudi previ és inviable. Necessitem, per tant, realitzar un anàlisi minuciós de la coordinació dels serveis d'emergència. La coordinació és molt important per aconseguir uns resultats satisfactoris i minimitzar el temps tant de resposta global com d'actuació de cada servei.

Aquest anàlisi es basa en diferents parts:

- Informació general: informació generada per l'accident, informació necessària per cada servei per dur a terme la intervenció.
- Coordinació de cada servei amb l'accident
- Coordinació entre els serveis

Per a realitzar aquests models que compleixen amb l'anàlisi i característiques esmentades es requereix l'ús de les **Xarxes de Petri**.

Per què usar-les per a la simulació de sistemes logístics?

Les Xarxes de Petri són una potent extensió dels grafs d'estats que proporcionen una representació més compacta i sobretot més efectiva dels sistemes concurrents.

Un dels seus punts més rellevants és l'aplicació en sistemes logístics gràcies sobretot a la seva simplicitat, potència i flexibilitat per simular aquests sistemes logístics (generalment complexes) d'una forma senzilla.

Aquesta aplicació pot ser de dues maneres diferents: mitjançant grafs bipartits directes (interpretació gràfica) o a partir d'equacions d'estats (interpretació matemàtica).

La seva gran popularitat en terrenys acadèmics i industrials ha permès desenvolupar una àmplia teoria matemàtica que permet l'anàlisi del comportament de sistemes que inclouen tècniques per a la seva optimització.

Aquestes característiques bàsiques i d'altres [DiCesare, 1993], són les raons per les quals les Xarxes de Petri (i sobretot les acolorides) són una eina ideal per a realitzar simulacions.

A continuació es realitzarà una breu explicació sobre les Xarxes de Petri ordinàries i més concretament sobre les Xarxes de Petri Acolorides.

2.2. Xarxes de Petri

2.2.1. Xarxes de Petri ordinàries

Les Xarxes de Petri (XdP) o Petri Nets (PN) són una eina molt útil i potent per a modelar i analitzar sistemes dinàmics basats en esdeveniments discrets. Ofereixen el formalisme necessari per representar tant els components que integren el sistema com la seqüència d'events que pot desencadenar d'un cert estat del sistema.

Són un cas particular d'un gràfic dirigit, ponderat i bipartit, amb dos tipus de nodes: transicions i llocs.

Nodes tipus transició (T): Representen els canvis d'estat.

Els representem gràficament per un rectangle i s'utilitzen per modelar esdeveniments que apareixen en la dinàmica del sistema.

Nodes tipus lloc (P): Representen l'estat del sistema.

Els representem gràficament per un cercle i s'utilitzen generalment tant per descriure les cues d'espera del sistema com per descriure condicions sobre l'estat sobre el que es troben els elements que integren el sistema.

Dins dels nodes lloc podem representar el número d'elements que hi ha a les cues d'espera mitjançant punts anomenats *tokens*.

Per tant, els llocs i les transicions configuren l'estructura de la xarxa i representen la part estàtica mentre que els tokens representen la part dinàmica del sistema.

En la figura 1 podem observar un node tipus lloc (a), una transició (b) i un node amb 3 tokens (c).



Figura 1: representació gràfica

Arcs (A): Els representem gràficament per fletxes i uneixen consecutivament una transició amb un lloc o viceversa. Acostumen a tenir un *pes* associat (**W**), el qual permet descriure les condicions necessàries per a que l'esdeveniment representat en la transició pugui activar-se.

Matemàticament una Xarxa de Petri la podem definir com una tupla de 5 elements:
XdP = (P,T,A,W,M₀) on

$$\begin{aligned}
 P &= \{P_1, P_2, \dots, P_n\} \\
 T &= \{T_1, T_2, \dots, T_m\} \\
 A &= (P \times T) \cup (T \times P) \\
 W &: A_i \mapsto \{1,2,3,\dots\} \quad \forall A_i \\
 M_0 &: P_i \mapsto \{1,2,3,\dots\} \quad \forall A_i
 \end{aligned}$$

Les regles per simular el comportament dinàmic d'una Xarxa de Petri (“regles de dispar”) són:

- * Una transició Ty està activada si cadascun dels nodes Pj tipus lloc connectats a l'entrada contenen almenys W (Pj , Ty) objectes tokens. On W (Pj , Ty) representa el pes de l'arc que uneix el node Pj amb la transició Ty. És a dir, està activada si tots els seus llocs d'entrada tenen com a mínim els mateixos tokens com marca el pes de l'arc.
 La figura 2 mostra una transició activada o habilitada (a) i una altra que no ho està (b).

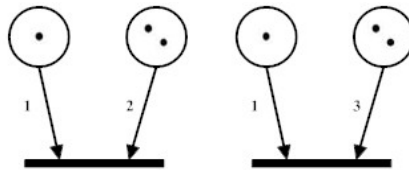


Figura 2: activació d'una transició

- * Una transició activada pot “disparar-se” en qualsevol moment.
- * Com a resultat de disparar una transició activada, s'eliminen W (Pj , Ty) objectes de cada node Pj a la entrada de Ty , i s'afegeixen W (Ty , Pk) objectes a cada node Pk de la sortida Ty. És a dir, el dispar d'una transició provoca que es tregui de cada lloc d'entrada un nombre de tokens igual al pes de l'arc de la connexió i vagi a parar al lloc de sortida complint la mateixa igualtat.

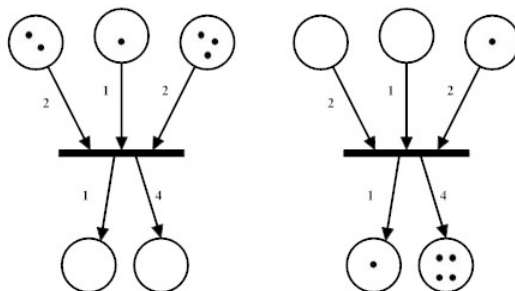


figura 3: resultat de disparar una transició

2.2.2. Xarxes de Petri Acolorides

Les Xarxes de Petri Acolorides (XdPA) o Coloured Petri Nets (CPN), són xarxes de Petri on els tokens poden tenir diferents atributs associats. Per atributs, entenem a tot el conjunt de valors que caracteritzen les entitats que flueixen en el sistema. Aquests atributs, poden ser de diferents tipus (enters, reals, caràcters, ...) i reben el nom de *colors*. Els colors, són els que donen el nom *acolorides* a aquest tipus de XdP en contrast a les normals on els tokens són definits com a punts sense colors (punts negres).

El nombre de tokens en cada node tipus lloc així com els valors dels colors associats a cada un dels tokens conformen el *vector d'estat*.

Les principals diferències respecte a les XdP clàssiques són:

1. Els nodes tipus llocs d'una XdPA poden tenir elements d'un tipus determinat.
2. Les transicions poden tenir *guardes*.
3. Els arcs d'una XdPA poden tenir *expressions d'arc*.

Avantatges de les XdPA:

Les Xarxes de Petri Acolorides tot i tenir la mateixa capacitat descriptiva que les Xarxes de Petri ordinàries tenen molta més capacitat de síntesis en el modelatge de certs tipus de sistemes com són els de producció o els logístics.

La seva capacitat de contenir tant l'estructura estàtica com la dinàmica del sistema així com la seva naturalesa gràfica les fan ideals per a la simulació.

Com ja hem comentat, el principal avantatge són la possibilitat dels tokens a agafar diferents valors de dades. Això, ens pot simplificar molt la feina alhora de dissenyar un sistema logístic o de producció on tenim diferents (serveis en el nostre cas) o màquines amb funcions similars. Per modelar-lo, no caldrà crear un submodel per a cada cas sinó que podrem definir-ne un de sol que les englobi totes.

Aquesta reducció de sistemes en "un de sol", ens permet generalitzar el model i per tant, aplicar-lo a models substancialment diferents (diferents nombres de serveis o màquines).

Són adequades per modelar i visualitzar patrons de comportament que mostren concurrència, sincronització i recursos compartits (factors claus per a la millora del rendiment en sistemes logístics o manufacturers).

Matemàticament la podem definir com una tupla de nou elements:

XdPA = ($\Sigma, P, T, A, N, C, G, E, I$) on

Σ : Conjunt finit i no buit de colors que permet especificar els colors que s'han de definir per cada tipus d'entitat que es vol modelar.

P: Conjunt finit de nodes tipus lloc que permeten especificar l'estat del sistema.

T: Conjunt finit de nodes tipus transició. Acostumen a representar inicis o finals d'una certa activitat o el final d'un succés extern.

A: Conjunt finit d'arcs.

N: Funció de node que permeten associar a cada arc els seus nodes terminals (node origen i no node destí).

C(Colors): Conjunt de funcions color que permeten especificar per cada node lloc el tipus d'entitats que poden guardar-se.

G(Guardes): Funcions booleanes associades a les transicions (es representen entre claudàtors). S'utilitza per desinhibir l'event associat a la transició en funció dels valors dels atributs de l'entitat que es vol processar. Una transició pot o no tenir-ne segons les necessitats.

E(Expressions): Expressions que s'associen als arcs que permeten especificar el tipus d'entitat del node lloc d'entrada a la transició que s'ha d'escollir d'entre totes per habilitar l'event. Si es troba associada a un arc de sortida de la transició, la expressió s'utilitza per avaluar els nous valors els atributs de les entitats de sortida.

I(Inicialització): Funció d'inicialització que permet especificar els valors dels colors de les entitats associades inicialment a un determinat lloc. S'expressa subratllant-lo.

Les regles o condicions de dispar d'una XdPA permeten restringir millor les condicions de dispar de les transicions respecte una XdP ordinària. L'ús de colors i d'expressions als arcs permeten a les XdPA obtenir un model més compacte que l'equivalent en XdP. Les condicions de dispar són :

- * Direm que una transició està activada si el guarda associat a ella s'avalua a CERT i en els seus llocs d'entrada hi ha el nombre de tokens indicats pels arcs d'entrada i aquests tenen els atributs especificats.
- * El dispar d'una transició, segueix el mateix procediment que la XdP convencional. El resultat de disparar una transició activada, es el canvi dels tokens indicats pels arcs d'entrada dels llocs d'entrada cap als llocs de sortida segons indiquin els arcs de sortida.

A continuació podem veure una XdPA que correspon a un sistema de producció de peces de diferents stocks mitjançant tres màquines diferents.

Exemple adaptat extret del llibre *Modelado y Simulación* [Guasch, et. al., 2003].

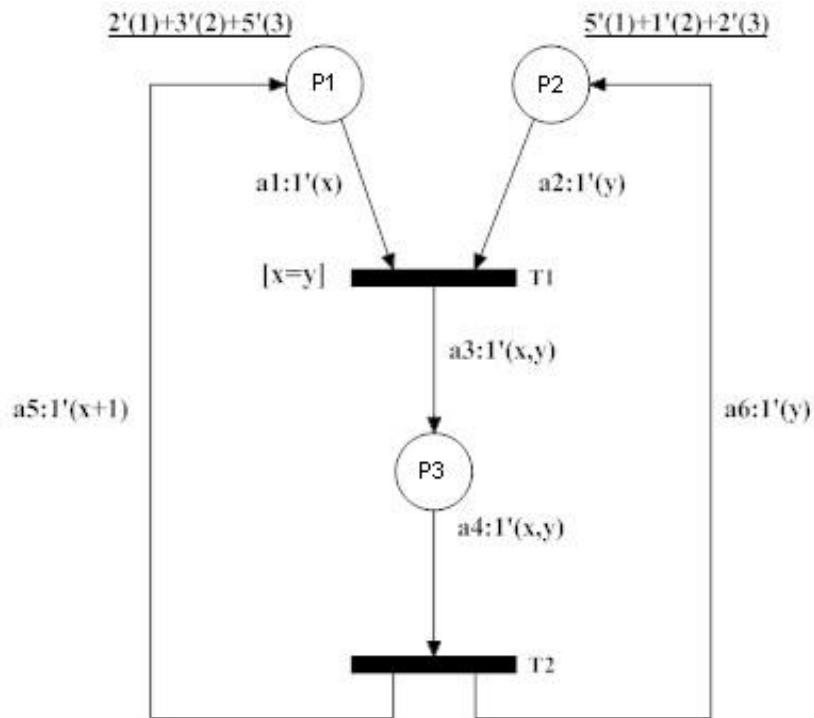


figura 4: Exemple d'una Xarxa de Petri Acolorida

Tenim tres tipus de màquines (definides pel color y) que produeixen peces agafant-les de quatre stocks diferents (cues d'espera definides amb el color x).

P1 correspon a aquelles peces pendents de processar en els diferents stocks mentre que P2 a les màquines que estan lliures.

Les expressions subratllades corresponen a la inicialització i ens determinen el número de màquines lliures de cada tipus i el número de peces acumulades en cada stock.

T1 ens marca l'inici d'aquest procés. Quan una màquina queda lliure i tenim una peça en stock iniciem el procés. Podem veure que tenim un guarda $[x=y]$ el qual ens diu que cada màquina processa les peces d'un determinat stock i no tots. Quan es compleixi la condició, s'activarà la transició i entrarem al procés (P3). T2 ens marcaria el final del procés de confecció d'una peça.

Seguidament tornariem a les condicions inicials alliberant la màquina utilitzada (arc6) i avançant una posició del stock (arc5) ja que la posició actual ha quedat lliure.

2.3. Aproximació al Model

2.3.1. Model Flow Chart

Una vegada adquirits els coneixements sobre les XdPA podem centrar-nos en el disseny del model. Abans, però, cal tenir clar l'esquema general del model.

Una primera aproximació del funcionament dels serveis i la coordinació entre aquests es mostra gràficament en el següent diagrama de flux.

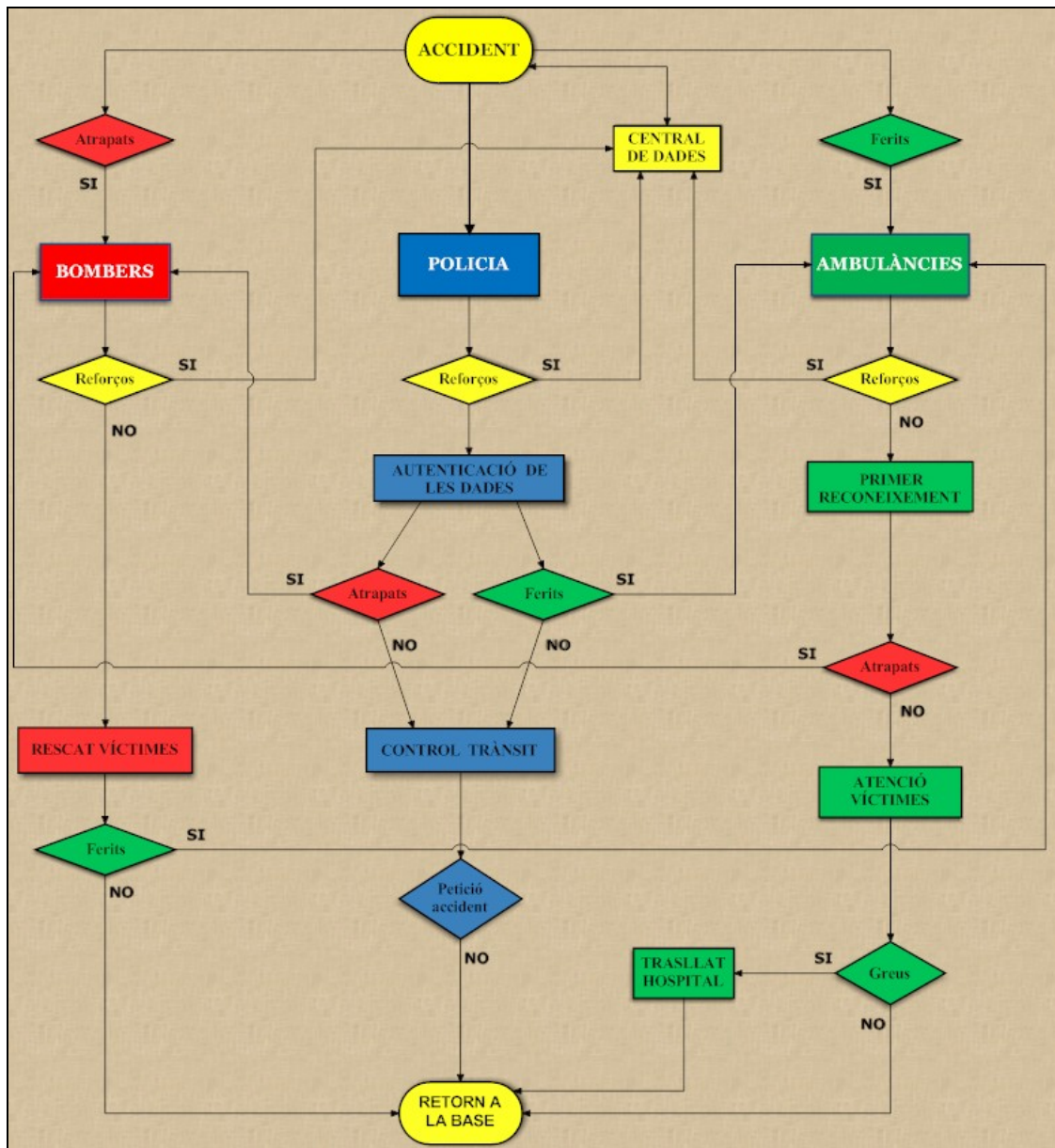


figura 5: Diagrama de flux: esquema general

Donada la complexitat alhora de dissenyar la Xarxa de Petri Acolorida (XdPA) es va optar primer per realitzar-les per separat. Les tres xarxes resultants corresponen als tres tipus de serveis d'emergència que podem tenir implicats en un accident: policia, bombers i ambulàncies.

Les tres xarxes presenten els següents nodes comuns: accident, vehicle i ruta (tot i que aquest últim només apareixerà en el submodel de moviment dels vehicles). Tots tres tenen connexió amb una base de dades central.

2.3.2. Central de Dades

La central de dades, és qui porta l'actualització de les dades sobre l'accident des de l'inici (té lloc l'accident) fins que és considera finalitzat. Aquesta actualització de la informació és constant i necessària en tot moment per a la coordinació dels serveis d'emergència.

El seu modus operandi al llarg d'un accident és el següent: quan es produeix un accident, la central rep el primer avís per via telefònica. Aquesta informació tot i ser vital per atendre l'emergència el més ràpid possible, sovint és poc fiable donat que el número de víctimes i sobretot el seu estat és canviant en el temps.

Així doncs, la central respon davant la primera alerta enviant un vehicle de reconeixement (policia). Una vegada aquest vehicle arribar al lloc de l'accident i és dona per acabat el reconeixement es considera que les dades ja són definitives (tot i que sempre poden aparèixer nous imprevistos). Seguidament, la tasca de la central consisteix en coordinar els serveis d'emergència (bombers i ambulància) assignant el nombre de vehicles necessaris cap al lloc de l'accident mantenint sempre la informació actualitzada a petició dels serveis.

La representació de la central de dades només es representarà en la XdPA general.

2.4. Definició dels nodes principals i els seus atributs o colors

Com s'ha comentat abans, les xarxes tenen tres nodes comuns: Accident (**A**), Vehicle (**V**) i Ruta (**R**).

Node Accident: Conté tota la informació relativa a l'accident (situació geogràfica, estat de l'accident, número de víctimes i vehicles involucrats en el sinistre, petició de reforços en cas de ser necessari...).

Node Vehicle: Conté tota la informació relativa a un vehicle determinat (tipus de vehicle, accident la qual ha estat assignat, estat del vehicle...).

Node Ruta : Conté tota la informació de la ruta utilitzada (posició d'inici i d'acabament, distància, temps...).

Els colors utilitzats per a definir-los són els següents:

Accident: A (ida, ea, la, evi, nf, nva, mv, nv, ap, ab, aa)

Color	Nom	Valors	Descripció – Serveis implicats
ida	identificador de l'accident	Enter	Identificador únic de cada accident. (Tots els serveis)
ea	estat de l'accident	1: pendent, 2: assignat, 3: en procés, 4: actualitzades, 5: finalitzat	Estats que marquen l'ordre en que es desenvolupa l'accident i la coordinació amb l'actuació dels serveis. (Tots els serveis)
la	localització de l'accident	Decimal	Posició geogràfica de l'accident. (Tots els serveis)
evi	estat de les víctimes	0: atrapades, 1: lleu, 2: greu	Estats que ens permeten avisar bombers (0) o ambulàncies (1 o 2) en cas d'activar-se. (Ambulàncies i bombers)
nf	número de ferits	Enter	Número total ja siguin lleus o greus. (Ambulàncies)
nva	número de víctimes atrapades	Enter	Número total de víctimes atrapades. (Bombers)
mv	més vehicles	0: no necessari, 1: necessari	Enviament de reforços en cas d'activar-se* (Tots els serveis)
nvi	número de vehicles implicats	Enter	Número total de vehicles involucrats en l'accident. (Policia)
ap	assignació policia	0: no assignat, 1: assignat	Flag de control assignació* (Policia)
ab	assignació bombers	0: no assignat, 1: assignat	Flag de control assignació* (Bombers)
aa	assignació ambulàncies	0: no assignat, 1: assignat	Flag de control assignació* (Ambulàncies)

* L'atribut **mv** així com els flags de control (**ap**, **ab** i **aa**) s'explicaran una vegada definides les tres XdPA bàsiques.

Node Vehicle: V(idv, ev, tv, idav, pv)

Color	Nom	Valors	Descripció – Serveis implicats
idv	identificador del vehicle	Enter	Identificador únic de cada vehicle. (Tots els serveis)
ev	estat del vehicle	1: lliure, 2: ocupat, 3: moviment	Estats possibles dels vehicles per a resoldre els accidents. (Tots els serveis)
tv	tipus vehicle	1: policia, 2: bombers, 3: ambulàncies	Posició geogràfica de l'accident. (Tots els serveis)
idav	identificador de l'accident assignat a un vehicle	Enter	Identificador únic dels vehicles que pertanyen a un mateix accident. (Tots els serveis)
pv	posició del vehicle	Decimal	Posició geogràfica d'un vehicle. (Tots els serveis)

Node Ruta: R(idr, pi, pf, pr, td)

Color	Nom	Valors	Descripció
idr	identificador de la ruta	Enter	Identificador únic de cada ruta
pi	posició inici ruta	Decimal	Posició inicial d'una ruta determinada
pf	posició final ruta	Enter	Posició final d'una ruta determinada
pr	posició dins la ruta	Enter	Posició actual d'una ruta determinada
td	temps a cobrir la distància	Decimal	Temps necessari per cobrir la distància

2.5. Disseny de les XdPA per separat

Una vegada especificats els nodes principals i la connexió amb la central de dades, es poden començar a dissenyar les tres xarxes bàsiques.

El disseny de cada xarxa s'ha dividit en tres parts essencials:

1. Informació sobre quines són les tasques principals i secundàries que ha de realitzar el servei en qüestió.
2. Declaració dels nodes, transicions i arcs necessaris.
3. Model de la Xarxa de Petri Acolorida (XdPA)

2.5.1. XdPA del cos de la Policia

1. Tasques que realitza:

El cos de policia té les següents funcions:

- És el primer vehicle enviat al lloc del sinistre encarregat d'informar a la central sobre les dades reals de l'accident: situació exacte, accés, número de vehicles involucrats, número de víctimes (lleus o greus) i víctimes atrapades.
- Una vegada acabat el reconeixement la seva funció rau en controlar el trànsit així com facilitar l'accés dels serveis d'emergència al lloc de l'accident.
- Quan les tasques es donen per acabades, l'accident està resolt. El vehicle de policia és l'únic que pot ser assignat a un altre accident directament sense tornar a la flota de vehicles. En cas contrari, el vehicle retorna a la flota.

2. Declaració dels nodes, transicions i arcs necessaris:

Nodes:

Els nodes utilitzats per a definir la XdPA (tant aquesta com les restants) és basen en els dos nodes generals (Accident i Vehicle). A part d'aquests dos, tenim altres nodes amb diferents noms i funcions, però conserven els mateixos colors (As i In en el cas d'accident i Va i V en el cas de vehicle).

Nom	Símbol	Descripció
Accident	A	Té lloc l'accident. Generació de la informació
Assignació	As	Assignació de l'accident al primer vehicle lliure de la flota
Vehicle accident	Va	Vehicle assignat en moviment cap al lloc de l'accident
Intervenció	In	Intervenció del servei de policia: primer reconeixement, control del trànsit i avis en cas de víctimes atrapades o ferits.
Vehicle	V	Vehicle en moviment cap a un altre accident o retornant a la flota

Transicions:

Les transicions són les accions que ens consumeixen temps. On es fan les accions. Necessitem realitzar sempre la comprovació $ida = idav$ en totes les transicions on s'actualitza l'accident, per aconseguir que s'actualitzi l'accident corresponent i no un qualsevol.

Núm.	Descripció	Guarda de les transicions
T1	Té lloc l'accident. Informació generada per l'accident.	--
T2	Accident pendent per assignar	--
T3	Flota de vehicles: vehicles disponibles per ser assignats	[ida=idav]
T4	Arribada al lloc de l'accident. Inici intervenció	[ida=idav]
T5	Fi intervenció	[ida=idav]
T6	Actualització estat moviment del vehicle	[ida=idav]

Arcs:

Els atributs que són números representen el valor concret que tenen en cada pas del model. Aquest valor es manté fins que no sigui modificat. Només cal especificar quin valor tenen en el moment de creació i quan s'actualitzen. Aquells atributs que conserven el seu nom signifiquen que el seu valor és indiferent o que encara no han estat inicialitzat.

Núm.	Expressió	Núm.	Expressió
a1	1'(ida, ea, la, evi, nf, nva, nvi)	a2	1'(ida, ea, la, evi, nf, nva, nvi)
a3	1'(ida, 1, la, evi, nf, nva, nvi)	a4	1'(ida, 1, la, evi, nf, nva, nvi)
a5	1'(ida, 1, la, evi, nf, nva, nvi)	a6	1'(ida, 2, la, evi, nf, nva, nvi)
a7	1'(ida, 3, la, evi, nf, nva, nvi)	a8	1'(idv, 3, 1, idav, pv)
a9	1'(idv, 3, 1, idav, pv)	a10	1'(ida, 4, la, evi, nf, nva, nvi)
a11	1'(ida, 3, la, evi, nf, nva, nvi)	a12	1'(idv, 1, 1, idav, pv)
a13	1'(idv, ev, 1, idav, pv)	a14	1'(ida, 3, la, evi, nf, nva, nvi)
a15	1'(ida, 3, la, evi, nf, nva, nvi)	a16	1'(ida, 5, la, evi, nf, nva, nvi)
a17	1'(ida, 5, la, evi, nf, nva, nvi)	a18	1'(idv, 2, 1, idav, pv)
a19	1'(idv, 3, 1, idav, pv)	a20	1'(idv, 2, 1, idav, pv)
a21	1'(idv, ev, 1, idav, pv)	a22	1'(idv, ev, 1, idav, pv)

2.5.2. XdPA del cos de Bombers

1. Tasques que realitza:

- El cos de bombers només actua una vegada ja disposem de les dades actualitzades sobre l'accident i en el cas que hi hagin víctimes atrapades així com petits incendis causats pels vehicles.
- El cos de bombers pot ser avisat per la central en primera instància, per la policia una vegada arribat al lloc del sinistre o pels serveis mèdics quan aquests han efectuat el primer reconeixement (opció més fiable).
- Una vegada acabat el servei, retornen a la flota.

2. Declaració dels nodes, transicions i arcs necessaris:

Nodes:

Els petits canvis respecte a la xarxa de policia estan relacionats amb el node vehicle. Els vehicles de bombers, una vegada acabada la intervenció tornen sempre a la base. Per aquesta raó afegim els nodes **Vf** i **VI** els quals representen el moviment cap a la flota i l'actualització de l'estat d'aquest a lliure.

Nom	Símbol	Descripció
Accident	A	Té lloc l'accident. Generació de la informació
Assignació	As	Assignació de l'accident al primer vehicle lliure de la flota
Vehicle accident	Va	Vehicle assignat en moviment cap al lloc de l'accident
Intervenció	In	Intervenció del servei de policia: primer reconeixement, control del trànsit i avis en cas de víctimes atrapades o ferits.
Vehicle flota	Vf	Vehicle en moviment cap a la flota
Vehicle lliure	VI	Vehicle lliure disponible a la flota

Transicions:

La transició **T6** canvia respecte al model anterior. Representa l'arribada del vehicle a la flota i el node següent representa que el vehicle torna a estar disponible (en disposició de ser assignat).

Novament, necessitem assegurar-nos que l'accident a actualitzar sigui el que volem usant el guardes de les transicions.

Nom	Descripció	Guarda de les transicions
T1	Té lloc l'accident. Informació generada per l'accident	--
T2	Accident pendent per assignar	--
T3	Flota de vehicles: vehicles disponibles per ser assignats	[ida=idav]
T4	Arribada al lloc de l'accident. Inici intervenció	[ida=idav]
T5	Fi intervenció. Retorn cap a la base	[ida=idav]
T6	Arribada a la base	--

Arcs:

Els arcs segueixen la mateixa estructura pels nodes relacionats amb l'accident (actualització del color **ea**) i vehicle (actualització **ev**), exceptuant el tipus vehicle que ara correspon al servei de bombers (**tv=2**).

Núm.	Expressió	Núm.	Expressió
a1	l'(ida, ea, la, evi, nf, nva, nvi)	a2	l'(ida, ea, la, evi, nf, nva, nvi)
a3	l'(ida, 1, la, evi, nf, nva, nvi)	a4	l'(ida, 1, la, evi, nf, nva, nvi)
a5	l'(ida, 1, la, evi, nf, nva, nvi)	a6	l'(ida, 2, la, evi, nf, nva, nvi)
a7	l'(ida, 3, la, evi, nf, nva, nvi)	a8	l'(idv, 3, 2, idav, pv)
a9	l'(idv, 3, 2, idav, pv)	a10	l'(ida, 4, la, evi, nf, nva, nvi)
a11	l'(ida, 3, la, evi, nf, nva, nvi)	a12	l'(ida, 3, la, evi, nf, nva, nvi)
a13	l'(ida, 3, la, evi, nf, nva, nvi)	a14	l'(ida, 5, la, evi, nf, nva, nvi)
a15	l'(ida, 5, la, evi, nf, nva, nvi)	a16	l'(idv, 3, 2, idav, pv)
a17	l'(idv, 3, 2, idav, pv)	a18	l'(idv, 1, 2, idav, pv)
a19	l'(idv, 1, 2, idav, pv)	a20	l'(idv, 1, 2, idav, pv)
a21	l'(idv, 1, 2, idav, pv)		

3. Model de la Xarxa de Petri Acolorida (XdPA)

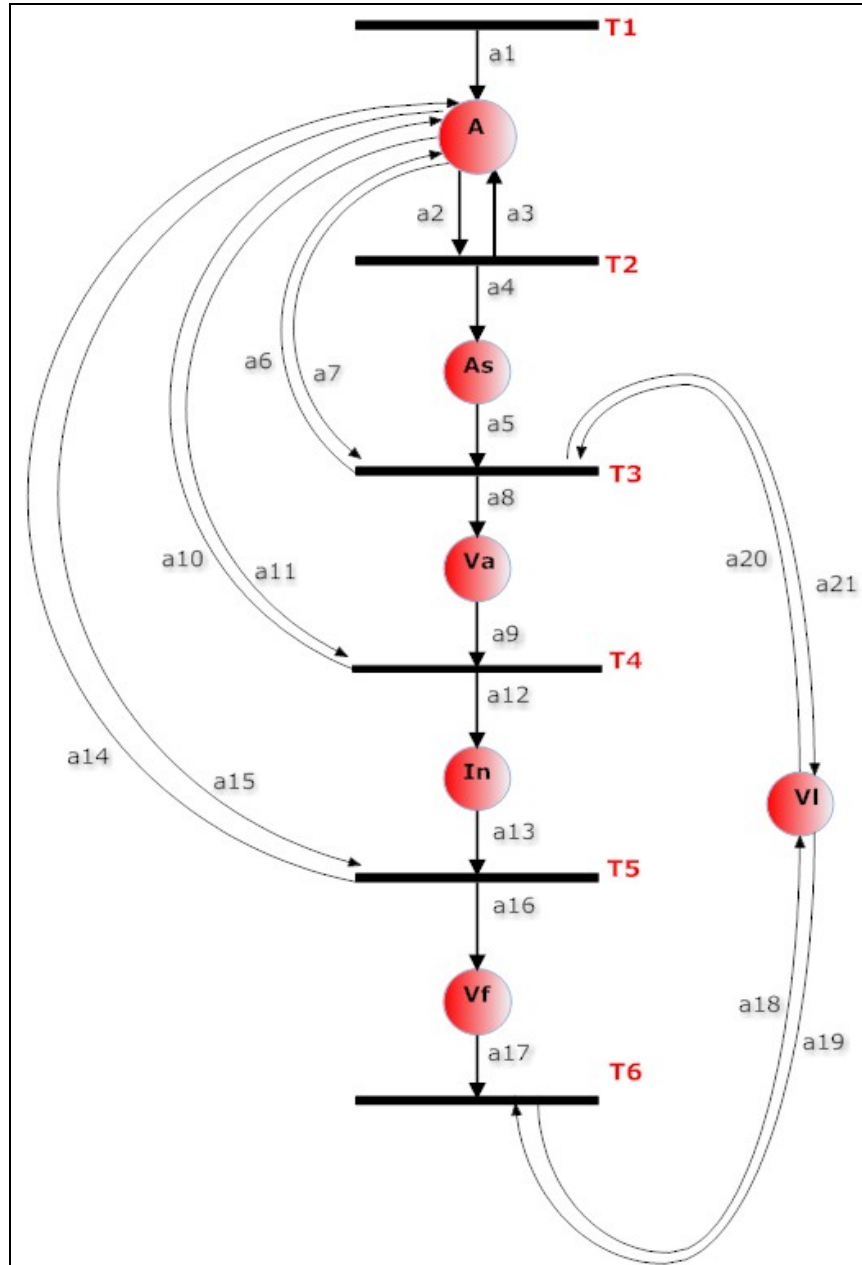


figura 7: XdPA Bombers

El node accident segueix el mateix patró que en la xarxa anterior, però en canvi, el node vehicle deixa d'estar centralitzat modificant-se en dos nodes (vf i vl) que mostra més clarament que els vehicles una vegada acabat el servei retornen a la flota i queden lliures per ser assignats altre cop.

2.5.3. XdPA del cos de les Ambulàncies

1. Tasques que realitza:

- Al igual que el cos de bombers, només actuen una vegada ja disposem de les dades reals sobre l'accident i només actuen si tenim víctimes ja siguin lleus o greus (atribut $evi=1$ o $evi=2$).
- La determinació de la gravetat de les víctimes es realitza en el lloc de l'accident. Les que es consideren greus (necessiten hospitalització) són les primeres que s'atenen i és realitza el trasllat cap a l'hospital.
- Si són lleus (no és necessària hospitalització) s'atenen allà mateix.
- Una vegada acabat el servei, retornen a la flota ja sigui des del lloc de l'accident o des de l'hospital.

2. Declaració dels nodes, transicions i arcs necessaris:

Nodes:

Mantenim els mateixos nodes que la xarxa de bombers amb la introducció d'un nou node (**Vh**) que simbolitza el trasllat i hospitalització de les víctimes considerades greus.

Nom	Símbol	Descripció
Accident	A	Té lloc l'accident. Generació de la informació
Assignació	As	Assignació de l'accident al primer vehicle lliure de la flota
Vehicle accident	Va	Vehicle assignat en moviment cap al lloc de l'accident
Intervenció	In	Intervenció del serveis mèdics: reconeixement a les víctimes (identificació de si són lleus o greus)
Vehicle hospital	Vh	Trasllat de les víctimes greus cap a l'hospital més proper.
Vehicle flota	Vf	Vehicle en moviment cap a la flota
Vehicle lliure	VI	Vehicle lliure disponible a la flota

Transicions:

La fi de la intervenció deriva en dues noves transicions: la que ens marca el retorn cap a la base i la que ens marca el trasllat cap a l'hospital. Apareix una nova transició (**T7**) conforme l'ambulància torna cap a la base des de l'hospital.

Novament, necessitem assegurar-nos que l'accident ha actualitzar sigui el que volem usant el guardes de les transicions.

Nom	Descripció	Guarda de les transicions
T1	Té lloc l'accident. Informació generada per l'accident	--
T2	Accident pendent per assignar	--
T3	Flota de vehicles: vehicles disponibles per ser assignats	[ida=idav]
T4	Arribada al lloc de l'accident. Inici intervenció	[ida=idav]
T5	Fi atenció víctimes greus. Trasllat cap a l'hospital	[ida=idav]
T6	Fi atenció víctimes lleus. Retorn cap a la base	[ida=idav]
T7	Arribada a l'hospital. Retorn cap a la base	--
T8	Arribada a la base	--

Arcs:

Núm.	Expressió	Núm.	Expressió
a1	l'(ida, ea, la, evi, nf, nva, nvi)	a2	l'(ida, ea, la, evi, nf, nva, nvi)
a3	l'(ida, 1, la, evi, nf, nva, nvi)	a4	l'(ida, 1, la, evi, nf, nva, nvi)
a5	l'(ida, 1, la, evi, nf, nva, nvi)	a6	l'(ida, 2, la, evi, nf, nva, nvi)
a7	l'(ida, 3, la, evi, nf, nva, nvi)	a8	l'(idv, 3, 3, idav, pv)
a9	l'(idv, 3, 3, idav, pv)	a10	l'(ida, 4, la, evi, nf, nva, nvi)
a11	l'(ida, 3, la, evi, nf, nva, nvi)	a12	l'(ida, 3, la, evi, nf, nva, nvi)
a13	l'(ida, 3, la, 2, nf, nva, nvi)	a14	l'(ida, 3, la, 1, nf, nva, nvi)
a15	l'(ida, 3, la, evi, nf, nva, nvi)	a16	l'(ida, 3, la, evi, nf, nva, nvi)
a17	l'(ida, 5, la, evi, nf, nva, nvi)	a18	l'(ida, 5, la, evi, nf, nva, nvi)
a19	l'(idv, 3, 3, idav, pv)	a20	l'(idv, 3, 3, idav, pv)
a21	l'(idv, 3, 3, idav, pv)	a22	l'(idv, 3, 3, idav, pv)
a23	l'(idv, 3, 3, idav, pv)	a24	l'(idv, 1, 3, idav, pv)
a25	l'(idv, 1, 3, idav, pv)	a26	l'(idv, 1, 3, idav, pv)
a27	l'(idv, 1, 3, idav, pv)		

Els arcs que ens determinen el camí a seguir una vegada començada la intervenció són **a13** i **a14** segons si es tracta de víctimes greus (2) o lleus (1). Si són lleus és dona el servei per acabat i tornem a la flota, mentre que si són greus el vehicle es dirigeix cap a l'hospital.

3. Model de la Xarxa de Petri Acolorida (XdPA)

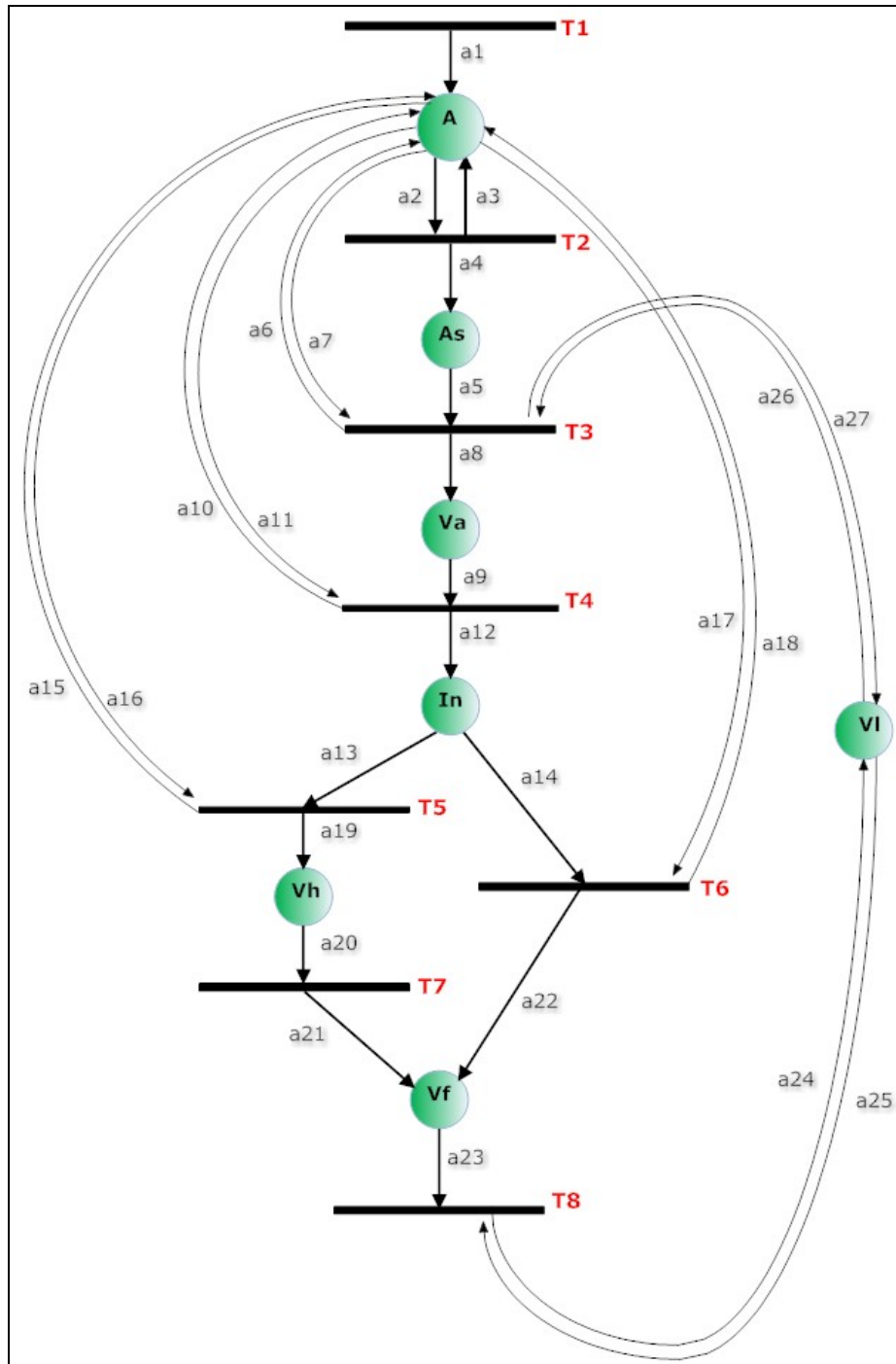


figura 8: XdPA Ambulàncies

Com ja s'ha comentat tenim dos transicions (**T5** i **T6**) que marquen la fi d'intervenció segons si tenim víctimes lleus (no cal fer res) o greus (cal traslladar-los a l'hospital). Es pot veure que independentment del cas, l'accident és dona per finalitzat en ambdós casos (arcs a15-16 i a17-18). Una vegada acabat segueixen la mateixa estructura que la xarxa de bombers (retorn a flota i vehicle lliure).

2.5.4. Submodel del moviment dels vehicles

1. Tasques que realitza:

Aquesta XdPA representa la relació d'actualització en el moviment dels vehicles respecte de la ruta triada. En cada moviment efectuat s'actualitza la posició del vehicle en ruta.

2. Declaració dels nodes, transicions i arcs necessaris:

Nodes:

Tenim dos nodes principals : Vehicle i Ruta.

Nom	Símbol	Descripció
Vehicle	V	Vehicles en moviment
Ruta	R	Ruta determinada

Transicions:

Per a poder actualitzar la posició, els identificadors (**idv** i **idr**) dels nodes **Vehicle** i **Ruta** han de coincidir. Per tant, **idv=idr** serà el guarda que activarà la transició **T1**.

Nom	Descripció	Guarda de les transicions
T1	Actualització de la posició	[idv=idr]

Arcs:

Als arcs només hem d'actualitzar la posició del vehicle: $pos = pos + 1$ però amb els colors definits (pr en el cas de la ruta i pv en el de vehicle).

Núm.	Expressió	Núm.	Expressió
a1	$1'(idv, ev, tv, idav, \mathbf{pv})$	a2	$1'(idv, ev, tv, idav, \mathbf{pv+1})$
a3	$1'(idr, pi, pf, \mathbf{pr}, td)$	a4	$1'(idr, pi, pf, \mathbf{pr+1}, td)$

3. Model de la Xarxa de Petri Acolorida (XdPA)

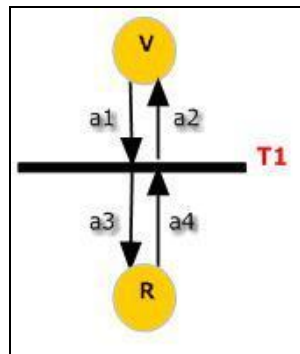


figura 9: XdPA Moviment

2.5.5. Actualització de les dades I: enviament de reforços i flags de control

Una vegada acabat el disseny dels tres models i el submodel bàsics, hem de considerar la següent situació: en un accident, es pot donar el cas en que una vegada actualitzades les dades (ja sigui per la policia o pels altres serveis quan arriben al lloc del sinistre), amb els vehicles enviats no en tinguem prou per a cobrir l'accident. És a dir, necessitem reforços.

Aquesta petició es fa a partir de l'atribut definit **mv** (node accident) el qual s'activa a 1 en el cas de que sigui necessari enviar més vehicles.

Entrar novament en l'assignació per assignar més vehicles en un mateix accident, ens obliga a definir uns flags de control. Aquests flags **ap**, **ab** i **aa** (un per a cada servei) seran els encarregats de només deixar-nos accedir a l'assignació quan necessitem reforços activant-se a 1 en cas de ser necessari.

Canvis realitzats:

En una transició, els colors d'entrada i sortida poden ser diferents. Fins i tot, es poden definir expressions condicionals, de manera que l'expressió d'arc de sortida vagi en funció de l'expressió de l'arc d'entrada. Aquestes expressions condicionals són les que ens permeten efectuar els canvis desitjats.

La petició la fa cada servei una vegada ha arribat al lloc de l'accident (transició **T4** de les tres xarxes). Si se'ns compleixen les condicions, l'arc **a10** és l'encarregat d'actualitzar les dades per enviar més vehicles i l'**a2** el que ens permet tornar a entrar a l'assignació.

A efectes pràctics s'ha considerat que per cada dues víctimes atrapades (nva) o ferides (nf) necessitem un vehicle ja sigui bombers o ambulàncies.

1. Policia: Necessita més efectius per cobrir el servei en funció dels vehicles involucrats en l'accident (**nvi**). Caldrà desactivar el flag **ap** per poder tornar a entrar per assignar més vehicles i activar el color **mv**.

Núm.	Condicció	Expressió
a10	si (ap=1 && nvi>2)	l'(ida, 4, la, evi, nf, nva, 1 , nvi, 0 , ab, aa)
a2	si (ap=0 (ap=0 && mv=1))	l'(ida, ea, la, evi, nf, nva, 0 , nvi, 1 , ab, aa)

2. Bombers: Necessitem més vehicles per atendre les víctimes atrapades (**nva>2**). Caldrà desactivar el flag **ab** per poder tornar a entrar per assignar més vehicles i activar el color **mv**.

Núm.	Condicció	Expressió
a10	si (ab=1 && nva>2)	l'(ida, 4, la, evi, nf, nva, 1 , nvi, 0 , ab, aa)
a2	si (ab=0 && (evi=0 nva>0) mv=1)	l'(ida, ea, la, evi, nf, nva, 0 , nvi, 1 , ab, aa)

3. Ambulàncies: Necessitem més vehicles per atendre les víctimes lleus o greus (**nf>2**). Necessitarem poder tornar a entrar per assignar més vehicles i activar el color **mv**.

Núm.	Condicció	Expressió
a10	si (nf>2 && aa=1)	l'(ida, 4, la, evi, nf, nva, 1 , nvi, 0 , ab, aa)
a2	si (aa=0 && (evi=1 evi=2 nf>0) mv=1)	l'(ida, ea, la, evi, nf, nva, 0 , nvi, 1 , ab, aa)

L'actualització de les dades no s'ha explicat fins ara degut a que és un punt una mica més complex i comú a les tres xarxes. S'ha optat primer fer una explicació detallada del funcionament bàsic de cada servei i després plantejar el problema de l'actualització. Així doncs, caldria afegir aquests quatre colors a tots els arcs de cada xarxa.

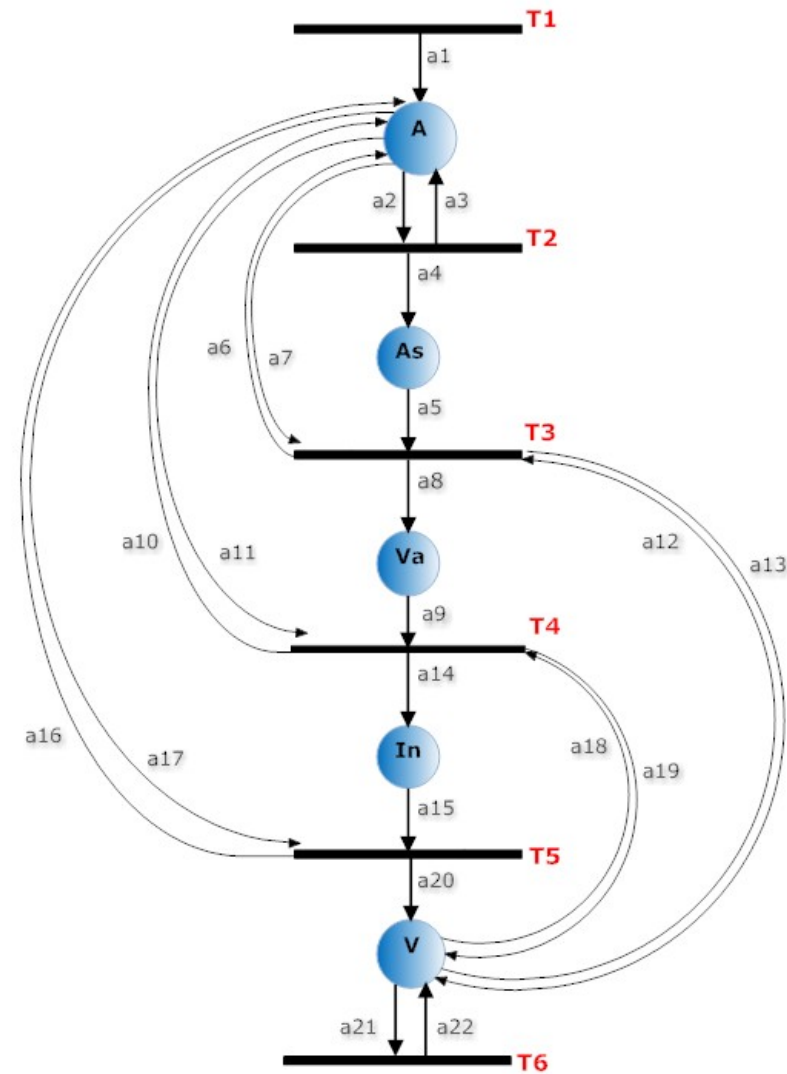
El disseny final de cadascuna de les tres xarxes de petri acolorides es mostren a continuació en format horitzontal per a una millor comprensió i visualització. Els canvis realitzats en els arcs es mostren en colors.

XdPA Policia:

- T1:** Té lloc l'accident
- T2:** Accident pendent per assignar
- T3:** Flota de vehicles: vehicles disponibles per ser assignats
- T4:** Arribada al lloc de l'accident. Inici intervenció
- T5:** Fi intervenció.
- T6:** Actualització estat moviment del vehicle

- A:** Accident
- As:** Assignació
- Va:** Vehicle en moviment cap al lloc de l'accident
- In:** Intervenció serveis
- V:** Vehicle en moviment cap a un altre accident o retornant a la flota

- a1:** 1'(ida, ea, la, evi, nf, nva, mv, nvi, 0, ab, aa)
- a2:** si (ap=0 || (ap=0 && mv=1)) → 1'(ida, ea, la, evi, nf, nva, 0, nvi, 1, ab, aa)
- a3:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a4:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a5:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a6:** 1'(ida, 2, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a7:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a8:** 1'(idv, 3, 1, idav, pv)
- a9:** 1'(idv, 3, 1, idav, pv)
- a10:** si (ap=1 && nvi>2) → 1'(ida, 4, la, evi, nf, nva, 1, nvi, 0, ab, aa)
- a11:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, 1, ab, aa)
- a12:** 1'(idv, 1, 1, idav, pv)
- a13:** 1'(idv, ev, 1, idav, pv)
- a14:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a15:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a16:** 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a17:** 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a18:** 1'(idv, 2, 1, idav, pv)
- a19:** 1'(idv, 3, 1, idav, pv)
- a20:** 1'(idv, 2, 1, idav, pv)
- a21:** 1'(idv, ev, 1, idav, pv)
- a22:** 1'(idv, ev, 1, idav, pv)

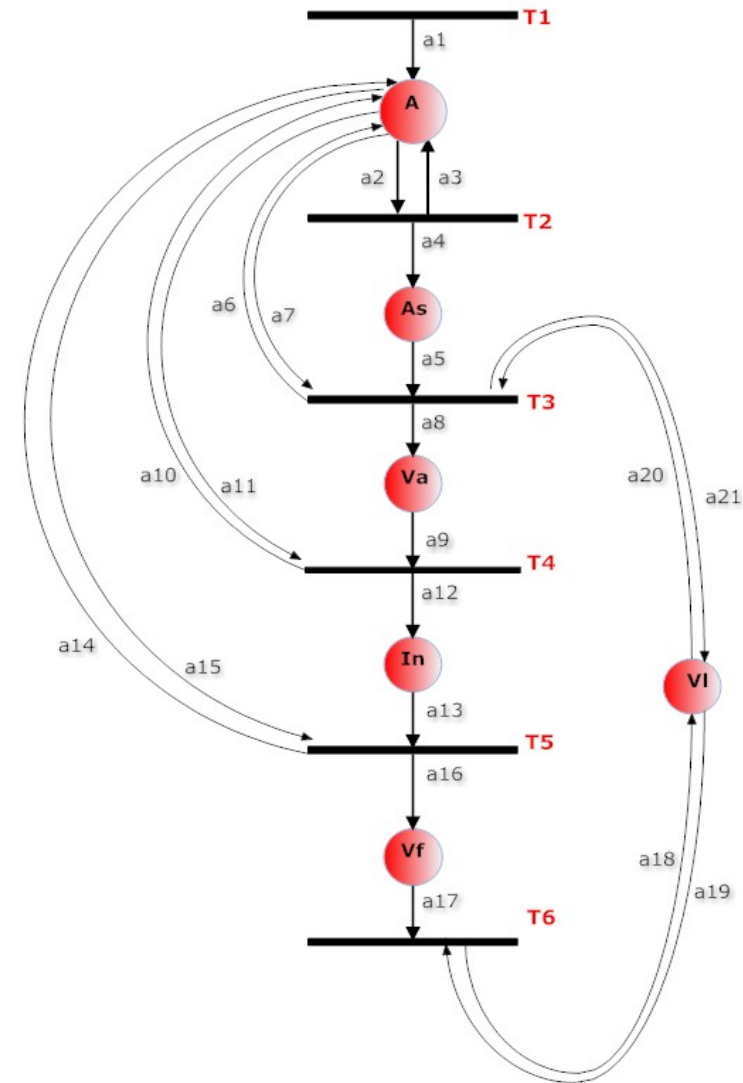


XdPA Bombers:

- T1:** Té lloc l'accident.
- T2:** Accident pendent per assignar
- T3:** Flota de vehicles: vehicles disponibles per ser assignats
- T4:** Arribada al lloc de l'accident. Inici intervenció
- T5:** Fi intervenció. Retorn cap a la base.
- T6:** Arribada a la base.

- A:** Accident
- As:** Assignació
- Va:** Vehicle en moviment cap al lloc de l'accident
- In:** Intervenció serveis
- Vf:** Vehicle en moviment cap a la flota
- VI:** Vehicle lliure disponible a la flota

- a1:** 1'(ida, ea, la, evi, nf, nva, mv, nvi, ap, 0, aa)
- a2:** si (ab=0 && (evi=0 || nva>0) || mv=1) → 1'(ida, ea, la, evi, nf, nva, 0, nvi, ap, 1, aa)
- a3:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a4:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a5:** 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a6:** 1'(ida, 2, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a7:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a8:** 1'(idv, 3, 2, idav, pv)
- a9:** 1'(idv, 3, 2, idav, pv)
- a10:** si (ab=1 && nva>2) → 1'(ida, 4, la, evi, nf, nva, 1, nv, ap, 0, aa)
- a11:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a12:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a13:** 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a14:** 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a15:** 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa)
- a16:** 1'(idv, 3, 2, idav, pv)
- a17:** 1'(idv, 3, 2, idav, pv)
- a18:** 1'(idv, 1, 2, idav, pv)
- a19:** 1'(idv, 1, 2, idav, pv)
- a20:** 1'(idv, 1, 2, idav, pv)
- a21:** 1'(idv, 1, 2, idav, pv)



2.6. XdPA General

El disseny de la xarxa general consisteix en la unió de les tres xarxes dels tres serveis. Per a la seva realització s'ha seguit el mateix patró que per les xarxes separades.

1. Primer disseny i unió de les tres xarxes:

Node Accident: Les tres xarxes tenen el seu punt d'unió en el node Accident (A). Cal tenir-lo centralitzat ja que la informació dels accidents per les tres xarxes és compartida. A efectes pràctics s'ha col·locat la xarxa de policia enmig donat que aquest servei està sempre present en un accident i per tant entrarem sempre.

Node Assignació: Necessitem 3 nodes d'assignació, ja que aquest procés es fa cada vegada que hi ha un canvi en les variables de l'accident que implica haver de recalculat els vehicles assignats. A més, l'assignació en cada servei es fa de forma independent en funció de les variables de l'accident.

Flotes de vehicles i actuacions dels serveis: La resta de nodes definits (flotes, moviments i intervencions) conservaran l'estructura de les tres xarxes per separat ja que en el fons, cada servei funciona independentment dels altres una vegada han rebut les peticions per intervenir.

Actualització de les dades: A diferència de les tres xarxes anteriors a les qual no s'ha mostrat la central de dades (encarregada de fer aquesta actualització de les dades de l'accident) a la xarxa general si que estarà representada (transició **T11**).

Finalització de l'accident: Una vegada la intervenció dels serveis d'emergència hagi acabat, s'actualitza l'estat de l'accident com a finalitzat ($ea=5$). Només caldrà fer-ho pel servei de policia ja que és l'únic que participa sempre en tots els accidents i és només una actualització d'estat, no de variables referents a l'accident.

2. Declaració dels nodes, transicions i arcs necessaris:

Com ja s'ha comentat, cada xarxa manté els seus nodes, transicions i arcs, a excepció de la transició **T11** (fins ara no mostrada) i els arcs referent a les assignacions.

Nodes:

Nom	Símbol	Descripció
Accident	A	Node central i compartit per les tres xarxes
Assignació	As	Assignació de cada servei
Vehicle accident	Va	Vehicle assignat en moviment cap al lloc de l'accident

Intervenció	In	Intervenció del servei de policia, bombers o ambulàncies
Vehicle hospital	Vh	Trasllat de les víctimes greus cap a l'hospital més proper.
Vehicle flota	Vf	Vehicle en moviment cap a la flota (bombers i ambulàncies)
Vehicle lliure	VI	Vehicle lliure disponible a la flota (bombers i ambulàncies)
Vehicle	V	Vehicle en moviment cap a un altre accident o retornant a la flota (policia)

Transicions:

Nom	Descripció	Guarda de les transicions
T1	Té lloc l'accident	--
T2	Accident pendent per assignar als bombers	--
T3	Accident pendent per assignar als serveis mèdics	[ida=idav]
T4	Accident pendent per assignar a la policia	[ida=idav]
T5	Assignació a la flota de vehicles del cos de bombers	[ida=idav]
T6	Assignació a la flota de vehicles del cos de policia	[ida=idav]
T7	Assignació a la flota de vehicles del cos d'ambulàncies	[ida=idav]
T8	Arribada al lloc de l'accident (bombers) Inici atenció víctimes atrapades	[ida=idav]
T9	Arribada al lloc de l'accident (policia) Inici control del trànsit	[ida=idav]
T10	Arribada al lloc de l'accident (ambulància) Inici atenció a les víctimes	[ida=idav]
T11	Central de dades (actualització dades accident)	--
T12	Fi intervenció (atenció víctimes atrapades) Retorn cap a la base	[ida=idav]
T13	Fi intervenció (control trànsit)	[ida=idav]
T14	Fi exploració víctimes greus. Inici trasllat cap a l'hospital	[ida=idav]
T15	Arribada a l'hospital (registre i ingrés) Retorn cap a la base	--
T16	Fi exploració víctimes lleus. Retorn cap a la base	[ida=idav]
T17	Vehicle en dipòsit	--
T18	Actualització moviment Vehicle	[ida=idav]
T19	Vehicle en dipòsit	--

Arcs:

Núm.	Expressió	Núm.	Expressió
a1	1'(ida, ea, la, evi, nf, nva, nvi)	a2	1'(ida, ea, la, evi, nf, nva, nvi)
a3	1'(ida, 1, la, evi, nf, nva, nvi)	a4	1'(ida, 1, la, evi, nf, nva, nvi)
a5	1'(ida, 1, la, evi, nf, nva, nvi)	a6	1'(ida, 2, la, evi, nf, nva, nvi)
a7	1'(ida, 3, la, evi, nf, nva, nvi)	a8	1'(ida, 1, la, evi, nf, nva, nvi)
a9	1'(ida, 1, la, evi, nf, nva, nvi)	a10	1'(ida, 1, la, evi, nf, nva, nvi)
a11	1'(ida, 1, la, evi, nf, nva, nvi)	a12	1'(ida, 1, la, evi, nf, nva, nvi)
a13	1'(ida, 1, la, evi, nf, nva, nvi)	a14	1'(ida, 2, la, evi, nf, nva, nvi)
a15	1'(ida, 3, la, evi, nf, nva, nvi)	a16	1'(ida, 2, la, evi, nf, nva, nvi)
a17	1'(ida, 3, la, evi, nf, nva, nvi)	a18	1'(ida, 2, la, evi, nf, nva, nvi)
a19	1'(ida, 3, la, evi, nf, nva, nvi)	a20	1'(idv, 3, 1, idav, pv)
a21	1'(idv, 3, 1, idav, pv)	a22	1'(idv, 1, 1, idav, pv)
a23	1'(idv, ev, 1, idav, pv)	a24	1'(idv, 3, 2, idav, pv)
a25	1'(idv, 3, 2, idav, pv)	a26	1'(idv, 3, 3, idav, pv)
a27	1'(idv, 3, 3, idav, pv)	a28	1'(ida, 4, la, evi, nf, nva, nvi)
a29	1'(ida, 3, la, evi, nf, nva, nvi)	a30	1'(ida, 4, la, evi, nf, nva, nvi)
a31	1'(ida, 3, la, evi, nf, nva, nvi)	a32	1'(ida, 4, la, evi, nf, nva, nvi)
a33	1'(ida, 3, la, evi, nf, nva, nvi)	a34	1'(ida, 4, la, evi, nf, nva, nvi)
a35	1'(ida, 3, la, evi, nf, nva, nvi)	A36	1'(idv, 2, 1, idav, pv)
a37	1'(idv, 3, 1, idav, pv)	A38	1'(ida, 3, la, evi, nf, nva, nvi)
a39	1'(ida, 3, la, evi, nf, nva, nvi)	a40	1'(ida, 3, la, evi, nf, nva, nvi)
a41	1'(ida, 3, la, evi, nf, nva, nvi)	a42	1'(ida, 3, la, evi, nf, nva, nvi)
a43	1'(ida, 3, la, 2, nf, nva, nvi)	a44	1'(ida, 3, la, 1, nf, nva, nvi)
a45	1'(ida, 5, la, evi, nf, nva, nvi)	a46	1'(ida, 5, la, evi, nf, nva, nvi)
a47	1'(ida, 5, la, evi, nf, nva, nvi)	a48	1'(ida, 5, la, evi, nf, nva, nvi)
a49	1'(ida, 5, la, evi, nf, nva, nvi)	a50	1'(ida, 5, la, evi, nf, nva, nvi)
a51	1'(ida, 5, la, evi, nf, nva, nvi)	a52	1'(ida, 5, la, evi, nf, nva, nvi)
a53	1'(idv, 3, 3, idav, pv)	a54	1'(idv, 3, 3, idav, pv)
a55	1'(idv, 3, 1, idav, pv)	a56	1'(idv, 3, 1, idav, pv)
a57	1'(idv, 2, 1, idav, pv)	a58	1'(idv, ev, 3, idav, pv)
a59	1'(idv, ev, 3, idav, pv)	a60	1'(idv, 3, 3, idav, pv)
a61	1'(idv, 3, 3, idav, pv)	a62	1'(idv, 3, 3, idav, pv)
a63,64	1'(idv, 1, 2, idav, pv)	a65,66	1'(idv, 1, 2, idav, pv)
a67, 68	1'(idv, 3, 1, idav, pv)	a69,70	1'(idv, 1, 3, idav, pv)

El disseny mostrat a la pàgina anterior ens mostra el funcionament dels tres serveis en conjunt. Tot i que ja s'ha explicat detalladament queda palès la centralització del node Accident per a les tres xarxes i la posterior divisió en les tres xarxes. S'ha inclòs en el disseny la central de dades (T11) i arcs a34-35 que simbolitzen la actualització de les dades.

2.6.1. Actualització de les dades II

L'actualització de les dades de l'accident, és necessària des de cada servei, ja que en realitat, la majoria de vegades que cal avisar als bombers ho fan directament els serveis mèdics (en una inspecció visual com la que fa la policia, a vegades és difícil determinar si hi han víctimes atrapades). Pot donar-se el cas que, en primera instància, només estigui activa la transició del procés de la policia, però que amb una actualització calgui activar ambulàncies o bombers.

Per tant, els canvis a realitzar són semblants als definits en les tres xarxes separades, però a part de comprovar si necessitem reforços per la xarxa en la que estem sinó que cal comprovar si s'han d'avisar als altres serveis.

Un altre punt que hem de considerar és bloquejar l'accés a aquell servei que no necessitem.

Aquest bloqueig és necessari ja que estalviem avaluar altres transicions (si no se'n activa el flag, no podem entrar a la xarxa corresponent) i per tant, temps i recursos.

Canvis realitzats:

Inicialment, els tres flags romandran a 0 (no necessaris)

arc 1: 1'(ida, ea, la, evi, nf, nva, 0, nv, 0, 0, 0)

1. Policia: Entrarem a l'assignació sempre (el servei de policia està sempre implicat en tots els accidents). Una vegada hem entrat, actualitzarem $ap = 1$ per no poder tornar a entrar. Si necessitem reforços (controlar trànsit) tornarem a desactivar el flag i enviarem reforços. Caldrà comprovar també si cal avisar bombers o ambulàncies (tenim víctimes atrapades o ferits).

Núm.	Petició	Condicció	Expressió
a6	Entrada	si (ap=0 (ap=0 && mv=1))	1'(ida, ea, la, evi, nf, nva, 0, nvi, 1, ab, aa)
a30	Reforços	si (ap=1 && nvi>2)	1'(ida, 4, la, evi, nf, nva, 1, nvi, 0, ab, aa)
	Bombers	si (evi=0 nva>0)	1'(ida, 4, la, evi, nf, nva, mv, nvi, ap, 0, aa)
	Ambulàncies	si (evi=1 evi=2 nf>0)	1'(ida, 4, la, evi, nf, nva, mv, nvi, ap, ab, 0)

2. Bombers: Entrarem a l'assignació de bombers ja sigui inicialment o si ens ha avisat la policia o els serveis mèdics. Només fan petició de reforços en cas de ser necessari.

Núm.	Petició	Condicció	Expressió
a2	Entrada	si (ab=0 && (evi=0 nva>0) mv=1)	l'(ida, ea, la, evi, nf, nva, 0 , nvi, ap, 1 , aa)
a28	Reforços	si (ab=1 && nva>2)	l'(ida, 4 , la, evi, nf, nva, 1 , nvi, ap, 0 , aa)

3. Ambulàncies: Si el que s'actualitza és el servei d'ambulàncies (aa=0), entrarem per primer cop a l'assignació d'ambulàncies. Quan arribin al lloc de l'accident, si necessiten més efectius, faran la petició per poder tornar a entrar a l'assignació. Les ambulàncies en el reconeixement de les víctimes poden avisar els bombers si ho consideren necessari.

Núm.	Petició	Condicció	Expressió
a4	Entrada	si aa=0 && (evi=1 evi=2 nf>0) mv=1)	l'(ida, ea, la, evi, nf, nva, 0 , nvi, ap, ab, 1)
a32	Reforços	si (nf>2 && aa=1)	l'(ida, 4 , la, evi, nf, nva, 1 , nvi, ap, ab, 0)
	Bombers	si (evi=0 nva>0)	l'(ida, 4 , la, evi, nf, nva, mv, nv, ap, 0 , aa)

Novament, aquests canvis afecten als arcs indicats, però en els arcs restants s'han d'incloure els colors **mv**, **ap**, **ab** i **aa**.

En la pàgina següents es mostra la versió definitiva dels arcs de la xarxa una vegada realitzats els canvis (mostrat amb els colors corresponents a cada xarxa).

Arcs XdPA Serveis d'emergència:

- | | | | |
|---------------------------------------------------------------------------|---|-----------------------------------------------------------------------|-----------------------------------------------------------------------|
| a1: 1'(ida, ea, la, evi, nf, nva, 0, nv, 0, 0, 0) | | | |
| a2: si (ab=0 && (evi=0 nva>0) mv=1) | → | 1'(ida, ea, la, evi, nf, nva, 0 , nvi, ap, 1 , aa) | a3: 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a4: si aa=0 && (evi=1 evi=2 nf>0) mv=1) | → | 1'(ida, ea, la, evi, nf, nva, 0 , nvi, ap, ab, 1) | a5: 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a6: si (ap=0 (ap=0 && mv=1)) | → | 1'(ida, ea, la, evi, nf, nva, 0 , nvi, 1 , ab, aa) | a7: 1'(ida, 1, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a8: 1'(ida, 1, la, evi, nf, nva, mv) | | a9: 1'(ida, 1, la, evi, nf, nva, mv) | a10: 1'(ida, 1, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a11: 1'(ida, 1, la, evi, nf, nva, mv) | | a12: 1'(ida, 1, la, evi, nf, nva, mv) | a13: 1'(ida, 1, la, evi, nf, nva, mv) |
| a14: 1'(ida, 2 , la, evi, nf, nva, mv, nv, ap, ab, aa) | | a15: 1'(ida, 3 , la, evi, nf, nva, mv, nvi, ap, ab, aa) | a16: 1'(ida, 2 , la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a17: 1'(ida, 3 , la, evi, nf, nva, mv, nv, ap, ab, aa) | | a18: 1'(ida, 2 , la, evi, nf, nva, mv, nvi, ap, ab, aa) | a19: 1'(ida, 3 , la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a20: 1'(idv, 3 , 1, idav, pv) | | a21: 1'(idv, 3 , 1, idav, pv) | a22: 1'(idv, 1 , 1, idav, pv) |
| a24: 1'(idv, 3 , 2, idav, pv) | | a25: 1'(idv, 3 , 2, idav, pv) | a23: 1'(idv, ev, 1 , idav, pv) |
| | | | a27: 1'(idv, 3 , 3, idav, pv) |
| a28: si (ab=1 && nva>2) | → | 1'(ida, 4, la, evi, nf, nva, 1 , nvi, ap, 0 , aa) | a29: 1'(ida, 3 , la, evi, nf, nva, mv) |
| a30: si (ap=1 && nvi>2) | → | 1'(ida, 4, la, evi, nf, nva, 1 , nvi, 0 , ab, aa) | a31: 1'(ida, 3 , la, evi, nf, nva, mv) |
| si (evi=0 nva>0) | → | 1'(ida, 4, la, evi, nf, nva, mv, nvi, ap, 0 , aa) | |
| si (evi=1 evi=2 nf>0) | → | 1'(ida, 4, la, evi, nf, nva, mv, nvi, ap, ab, 0) | |
| a32: si (nf>2 && aa=1) | → | 1'(ida, 4, la, evi, nf, nva, 1 , nvi, ap, ab, 0) | a33: 1'(ida, 3 , la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| si (evi=0 nva>0) | → | 1'(ida, 4, la, evi, nf, nva, mv, nv, ap, 0 , aa) | |
| a34: 1'(ida, 4, la, evi, nf, nva, mv, nvi, ap, ab, aa) | | a35: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) | <i>Acualització dades</i> |
| a36: 1'(idv, 2, 1, idav, pv) | | a37: 1'(idv, 3, 1, idav, pv) | a40: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a38: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) | | a39: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) | a43: 1'(ida, 3, la, 2, nf, nva, mv, nvi, ap, ab, aa) |
| a41: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) | | a42: 1'(ida, 3, la, evi, nf, nva, mv, nvi, ap, ab, aa) | a46: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a44: 1'(ida, 3, la, 1, nf, nva, mv, nvi, ap, ab, aa) | | a45: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) | a49: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a47: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) | | a48: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) | a52: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) |
| a50: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) | | a51: 1'(ida, 5, la, evi, nf, nva, mv, nvi, ap, ab, aa) | |
| a53: 1'(idv, 3, 3, idav, pv) | | a54: 1'(idv, 3, 3, idav, pv) | a59: 1'(idv, ev, 3, idav, pv) |
| a55: 1'(idv, 3, 1, idav, pv) | | a56: 1'(idv, 3, 1, idav, pv) | a62: 1'(idv, 3, 3, idav, pv) |
| a57: 1'(idv, 2, 1, idav, pv) | | a58: 1'(idv, ev, 3, idav, pv) | |
| a60: 1'(idv, 3, 3, idav, pv) | | a61: 1'(idv, 3, 3, idav, pv) | |
| a63: 1'(idv, 1, 2, idav, pv) | | a64: 1'(idv, 1, 2, idav, pv) | a65: 1'(idv, 1, 2, idav, pv) |
| a67: 1'(idv, 1, 3, idav, pv) | | a68: 1'(idv, 1, 3, idav, pv) | a66: 1'(idv, 1, 2, idav, pv) |
| | | | a70: 1'(idv, 1, 3, idav, pv) |

2.7. Simplificació de la XdPA General

Aprofitant les característiques de les XdPA (capacitat de síntesis i generalització) podem finalment fer una simplificació del model reduint-lo al màxim. Aquesta reducció, permet una representació final del model gràfic molt senzilla (tres nodes i dues transicions), però una complexitat elevada. Necessitarem definir nous colors per aconseguir la representació final del moviment així com nous guardes a les transicions. El model resultant, per tant, és menys intuïtiu i difícil de comprendre que els anteriors i per aquest motiu, la simplificació ha quedat com a punt final d'aquesta representació.

A més, aquest model ens apropa més al funcionament del moviment dels vehicles i posicions de les flotes que realitzarem en el model de simulació VBA.

1. Declaració dels nodes, transicions i arcs necessaris:

Nodes:

Els nodes, passen a ser generalitzats (Accident, Vehicle i Ruta) mantenint les mateixes accions que els definits en el model anterior, però en un de sol.

Nom	Símbol	Descripció
Accident	A	Informació de l'accident
Vehicle	V	Informació dels vehicles i moviment
Ruta	R	Informació de la ruta a seguir

Colors:

Els nous colors a definir afecten als nodes **Accident** i **Vehicle**. Tenen com a referència la localització de l'hospital i la flota dels vehicles. Gràcies a la seva definició, aconseguim una representació completa envers el moviment i la representació.

Color	Nom	Valors	Descripció
nodhos	Node hospital	Enter	Situació de l'hospital
flotv	Flota vehicle	Enter	Situació de la flota de vehicles

Transicions:

Pel que fa a les transicions, la primera (T1) segueix marcant la generació d'un accident mentre que la segona (T) engloba quatre transicions (informació i assignació de recursos a l'accident, moviment de vehicles i operació en el lloc de l'accident i hospitalització). Cadascuna, tindrà els guardes corresponents per disparar-les segons convingui.

Nom		Descripció	Guarda de les transicions
T1		Informació generada per l'accident	--
T	T2	Informació generada des del lloc de l'accident	[ida=idav] && {[flotv=pi] && [la=pf]} {[la=pi] && [nodhos=pf]} {[nodhos=pi] && [flotv=pf]} {[tv=1]&&[pv=pi] &&[la=pf]}
T	T3	Assignació de recursos a l'accident	
T	T4	Moviment dels vehicles	
T	T5	Actuació del servei / Hospitalització	

Arcs:

A excepció de l'arc a1, la resta, són bidireccionals respecte T. Això ens permet tenir en tot moment la informació i actualització de les accions realitzades pels tres nodes quan es disparen les transicions segons els guardes.

Núm.	Expressió
a1	1'(ida, ea, la, evi, nf, nva, nvi,nodhos)
a2	1'(ida, ea, la, evi, nf, nva, nvi,nodhos)
a3	1'(ida, ea, la, evi, nf, nva, nvi,nodhos)
a4	1'(idr, pi, pf, pr, td)
a5	1'(idr, pi, pf, pr, td)
a6	1'(idv, ev, tv, idav, pv, flotv)
a7	1'(idv, ev, tv, idav, pv, flotv)

2. Model de la Xarxa de Petri Acolorida (XdPA)

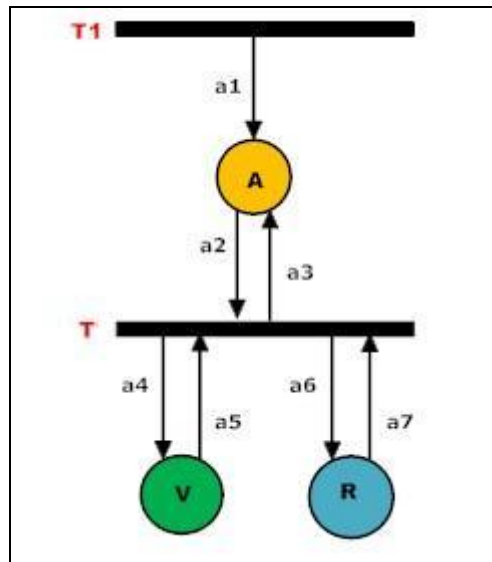


figura 11: XdPA simplificada

3. SIMULACIÓ EN ARENA

3.1. Què entenem per simulació?

La simulació digital és una tècnica que permet imitar (o simular) en un ordinador, el comportament d'un sistema¹ físic o teòric a partir d'unes condicions particulars d'operació.

Un projecte de simulació acostuma a seguir un ordre establert dividit en etapes.

3.1.1. Etapes d'un projecte de simulació

- **Formulació del problema:** Problema a estudiar, enumeració dels objectius.
- **Disseny del model conceptual:** Especificació del model a partir de les característiques dels elements del sistema que es volen estudiar i les seves interaccions tenint en compte els objectius del problema.
- **Recollida de dades:** Identificar, recollir i analitzar les dades necessàries per a l'estudi.
- **Construcció del model:** construcció del model de simulació a partir del model conceptual i el de dades.
- **Verificació i validació:** Comprovar que el model es comporta de la manera desitjada i existeix la correspondència adequada entre el sistema real i el model.
- **Anàlisi:** analitzar els resultats de la simulació amb la finalitat de detectar problemes i anomenar millores o solucions.
- **Documentació:** Proporcionar documentació del treball elaborat.
- **Implementació:** Posar en pràctica les decisions preses amb l'ajut de l'estudi de simulació.

¹ En l'annex A es pot trobar una descripció detallada sobre que entenem per un sistema, per un model i una classificació d'aquests.

3.1.2. Avantatges i inconvenients d'una simulació

Avantatges:

- Permet augmentar el coneixement sobre la dinàmica del procés que s'està estudiant gràcies a poder realitzar infinitat de proves.
- Avaluar la sensibilitat dels paràmetres del sistema.
- Permet analitzar l'efecte sobre el rendiment global d'un sistema, de petits canvis realitzats en una o algunes de les seves components.
- Permet experimentar en condicions en que l'experimentació amb el sistema real podria ser perillosa o d'un elevat cost econòmic.
- Estalvi econòmic i de temps: permet preveure tots aquells comportaments o resultats no desitjats que generen errors provocant pèrdues econòmiques i de temps.
- Optimitzar el comportament del sistema.

Inconvenients:

- Les solucions acostumen a no ser exactes ja que els resultats numèrics obtinguts es basen en un conjunt específic de nombres aleatoris.
- Alt cost: per obtenir estimacions més exactes, s'han de realitzar un gran nombre de rèpliques per a cada simulació. Això suposa una major dedicació i per tant més diners.
- Aprenentatge: tot i que és un factor canviant avui en dia, alguns programes de simulació no són fàcils d'utilitzar. Es necessita un aprenentatge tan per usar-los com per interpretar correctament els resultats.
- Risc de prendre decisions errònies basades en altres models de simulació que no han estat validats o verificats adequadament.

3.1.3. Aplicacions

La simulació actualment la trobem aplicada a una gran diversitat de camps dels quals destaquen:

- Processos de fabricació o producció (un dels primers camps en els quals es va usar la simulació).
- Logística: Contribueix de forma significativa la millora dels processos logístics en general.
- Transport: Part derivada de l'anterior usat per la majoria d'empreses relacionades amb la logística per coordinar i sobretot millorar el temps d'entrega o abastament dels seus productes.
- Sanitat: usats tant en la coordinació dels serveis mèdics d'una regió com per la logística associada per als trasplantaments i fins i tot per a la millora d'un centre hospitalari.
- Negocis: simulacions en les empreses dels processos tant administratius i de negocis com de control.
- Serveis: públics, bancs, empreses diverses, ...

3.1.4. Llenguatges de programació

Podem trobar molts tipus de llenguatges de programació per a realitzar simulacions. La gràcia d'aquests llenguatges és que l'usuari no ha de centrar-se exclusivament en la programació de la simulació sinó en la descripció del sistema.

Aquesta descripció es realitza amb un seguit d'eines que ofereixen la majoria de llenguatges i posteriorment aquesta descripció es converteix en un programa d'ordinador que serà l'encarregat d'executar la simulació.

En el mercat podem trobar SW de simulació destinat a simular camps específics i d'altres amb un àmbit més general que permeten simular qualsevol sistema (com el que farem servir i descriurem a continuació).

3.2 Què és Arena?

Arena és un simulador de propòsit general en el que l'entorn visual i la animació gaudeixen de molta importància. Aquestes característiques permeten apropar el món de la simulació a molta més gent i facilitar-los el procés d'elaboració, validació i experimentació gràcies a les eines gràfiques. Fins ara, això era incompatible amb un programa de simulació (ús excessiu de memòria i computacionalment intensiu), però la major capacitat de processament dels ordinadors d'avui en dia han solucionat aquest inconvenient.

El llenguatge de simulació usat pel SW Rockwell Arena és el **SIMAN**.

D'altres característiques [Kelton, et. al, 1998] que podem extreure del programa Arena són:

- Facilitat d'ús, flexibilitat i capacitat de modelatge.
- Proporciona la tecnologia necessària per completar amb èxit totes les fases d'un estudi de simulació.
- Estalvi d'errors causats per decisions basades únicament en la intuïció.
- Dissenyar processos que responen adequadament a la incertesa associada a la variabilitat i aleatorietat que afecten al sistema.
- Optimització: Incrementa la eficiència i la efectivitat d'un sistema existent.
- Elimina els colls de botella millorant l'ús dels actius existents.
- Millora el servei d'atenció al client mitjançant un bon servei i subministrament.
- Integració i anàlisi de dades d'entrada(integració amb Excel, Acces, ...).
- Visualització i animació dels processos: permet la importació des d'AutoCAD i connexió amb SW de Realitat Virtual amb Arena VRABox.
- Excel·lents informes i eines per anàlisi de resultats (exportació d'informes a Word, Excel, HTML,...).
- Arena combina les facilitats d'ús d'altres simuladors d'alt nivell amb la flexibilitat dels llenguatges de simulació i la possibilitat d'incorporar llenguatges com C o Visual Basic.
- L'entorn gràfic així com els menús del programa són força intuïtius i fàcils d'assimilar.

Per a realitzar la simulació s'ha usat la versió 7.0. L'Arena permet construir el model que volem dissenyar mitjançant un conjunt d'elements i una lògica del sistema. El resultat obtingut és un diagrama de blocs que integra tota la informació del model i mitjançant el qual és possible realitzar les simulacions necessàries.

Aquest conjunt d'elements i lògica del sistema el conformen:

Entitats: Són l'element dinàmic de les simulacions. Aquestes, segueixen el següent procés: es creen, es mouen a través del model canviant d'estat i interactuant amb d'altres entitats i/o el sistema, i finalment l'abandonen o són destruïdes.

En la nostra simulació tindrem dos tipus d'entitats: accident i vehicle.

Atributs: Els atributs són les característiques de les entitats (serien l'equivalent als colors de les XdPA). Els atributs poden tenir un seguit de valors de diferents tipus (enters, booleans, decimals, ..) i un valor específic agafat per una entitat és el que la fa diferenciar de les altres. El programa Arena té alguns atributs definits pels quals es pot fer un seguiment manera automàtica, però en la majoria de sistemes que vulguem simular serà necessari definir-ne de nous, assigna'ls-hi valors, modificar-los, etc.

Variables: una variable és una informació concreta que reflexa una característica del sistema. En Arena totes les variables es consideren globals i són independents de les entitats declarades. Tot i així, les entitats poden variar el valor de les variables agafant el seu valor o modificant-lo. Al igual que els atributs, tenim variables ja declarades en Arena. Un exemple són el temps de simulació, el número d'unitats d'una cua o el número d'unitats ocupades d'un recurs. L'altre tipus de variables són aquelles definides per l'usuari.

Recursos: Les entitats competeixen per entrar o obtenir recursos. En cas de no poder accedir a un recurs (ocupat per una altra entitat), les entitats s'han d'esperar a que el recurs sigui alliberat. Un exemple molt clar és el d'una màquina ocupada o l'espai limitat en un magatzem. Una entitat pot rebre simultàniament servei de diferents recursos.

Cues: Quan una entitat no pot continuar el seu moviment a través del model (normalment perquè no pot accedir a un recurs que està ocupat), es necessita un espai on fer esperar l'entitat fins que el recurs quedi lliure. Aquest espai el conformen les cues. Les cues poden tenir una mida limitada o no segons interressi.

Acumulador d'estadístiques: Arena permet catalogar en diferents grups les estadístiques dels resultats obtinguts en la simulació. En són exemples el número total de peces produïdes o el temps total de la simulació o el cost total d'aquesta. També podem accedir a una entitat, cua, recurs o variable concreta.

Events: Successos que ocorren en un moment determinat i poden canviar les entitats, els atributs, les variables i per tant els resultats obtinguts. En són exemples en el nostre disseny l'arribada d'un accident o la d'un vehicle al sistema.

Com ja s'ha comentat en les característiques del SW Arena, té un pes important la possibilitat de construir un *model visual*. La gran avantatge d'aquest model és que es pot observar en tot moment la simulació del procés.

3.3 Construcció del model

Per a construir el model són necessaris l'ús de *mòduls* o *blocs* dels quals es compon el programa. Aquests *mòduls* estan classificats en tres panells segons la seva funció. Aquests tres panells són: *Basic Process Panel*, *Advanced Process Panel* i *Advanced Transfer Panel*².

Per a realitzar el model s'ha seguit la mateixa estructura que la XdPC General no simplificada o estesa. S'ha usat aquesta i no la simplificada ja que de moment ens interessa poder distingir clarament els tres serveis. Les analogies amb la xarxa vénen donades per intentar substituir cada node junt amb el canvi d'estat (ja sigui de l'accident o del vehicle corresponent) amb una línia de simulació. A més, cada color existent en els nodes definits es correspon a un atribut en el sistema.

El resultat, per tant, és un model força intuïtiu i comprensible una vegada dissenyades les XdPA.

3.3.1. Línies d'acció: Consideracions prèvies

Com ja s'ha comentat, cada línia de simulació té una funció concreta i és comuna als tres serveis. Totes elles presenten una estructura i funcionament similar.

La idea general, es basa en crear entitats accidents i entitats vehicles les quals guardem en unes cues (accidents pendents i flotes de vehicles respectivament).

Entitats accident: Són les que inicien i finalitzen el procés. Per fer-ho, generem rèpliques de les originals (guardades a la cua) les quals aniran recorrent totes les línies d'acció comentades anteriorment (assignació, moviment, intervenció,...). Les originals, només les usarem per actualitzar l'estat d'un accident quan sigui necessari.

Entitats vehicle: Intervenien en el procés una vegada se'ls hi assigna un accident. Només treballarem amb entitats originals i també necessitem actualitzar l'estat dels vehicles en certs moments del procés.

Aquestes actualitzacions mencionades (estats dels accidents i dels vehicles) són necessàries perquè les entitats evolucionin dins el model canviant de línia d'acció amb els valors desitjats.

² Es pot trobar una descripció detallada dels blocs que conformen cada panell en la documentació adjunta (annex B.2).

Tot seguit es farà una descripció detallada de cada línia de procés:

3.3.2. Creació d'accidents

Línia encarregada de crear els accidents i posar en marxa el sistema.

Create: La creació de les entitats accidents és fa mitjançant el mòdul *create* usant una funció de probabilitat per generar accidents aleatòriament. Concretament la que usarem és una exponencial: *EXPO(25)*.

Assign: Fem l'assignació de tots els atributs que conformen els colors de l'Accident i les variables necessàries. Per a la declaració dels atributs s'han usat les següents distribucions de probabilitat³:

```

Attribute, nf, ANINT(UNIF(0,12))
Attribute, nvi, ANINT(UNIF(0,5))
Attribute, nva, ANINT(UNIF(0,8))
Attribute, evi, ANINT(TRIA(1,1,2))
Attribute, ida, Entity.SerialNumber
Attribute, ea, 1

```

Duplicate: El mòdul *duplicate* ens servirà per començar el procés enviant una rèplica de l'accident creat cap a cada servei. L'entitat original anirà a parar a la cua d'accidents (*hold*).

Hold: Representa la cua d'accidents pendents que arriben del *create*. Una vegada creades, les entitats de la cua s'aniran actualitzant (atribut *ea*) en cada línia de procés on intervingui l'accident.

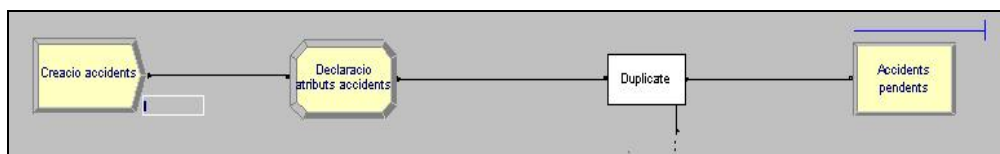


Figura 12: creació d'accidents

³ Arena permet crear un gran nombre de funcions de distribució de probabilitats gràcies al *Builder Expression*. Per més informació veure annex B.3.

3.3.3. Creació de les flotes de vehicles

Línia de creació de les bases dels vehicles comuna a les tres flotes. Usarem els mateixos mòduls que per crear accidents.

Create: La diferència amb el create anterior és que aquí no usarem una funció de probabilitat sinó que farem servir el valor constant 0 ja que ens interessa disposar de tots els vehicles abans no arribi el primer accident.

Assign: Novament declararem tots els atributs dels Vehicles i els enviarem a la cua de vehicles (*hold*). La declaració dels atributs és:

```

Attribute, idv, Entity.SerialNumber
Attribute, idav, 0
Attribute, tv, ANINT(TRIA(1,3,3))
Attribute, ev, 1
    
```

Hold: La cua Vpendents representa les flotes dels vehicles dels tres serveis. Novament aquells vehicles que entrin en servei s'aniran actualitzant retornant a la cua de vehicles (atribut ev).

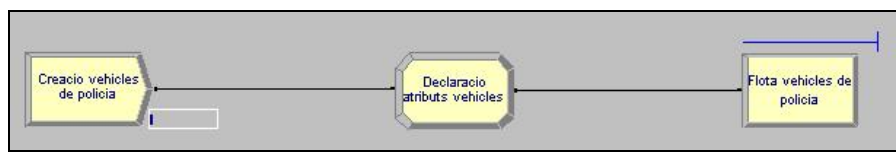


Figura 13: creació de les flotes de vehicles

Una vegada disposem d'accidents i de vehicles podem començar el procés. La primera línia d'acció seria la d'assignació, però prèviament necessitem saber en quina dels tres serveis volem entrar. La manera de fer-ho serà a partir de les condicions que havíem definit en les XdPA. Aquestes condicions les farem amb el mòdul *decide*.

3.3.4. Entrada línies d'acció: mòdul Decide

Aquest mòdul situat davant de les línies d'assignació serà el que ens permetrà l'entrada en el servei si és compleixen unes condicions o en cas contrari bloquejarà l'entrada. Les condicions per a cada servei són les següents:

Policia: condició: $(NQ(Vpendents) > 0)$.

Bombers: condició: $NQ(vpdependents) > 0 \ \&\& \ nva > 0$.

Ambulàncies: $NQ(vpdependents) > 0 \ \&\& \ nf > 0$.

3.3.5. Línia d'assignació

Aquesta línia combina les dues entitats (accident i vehicle) per a realitzar l'assignació d'un accident a un vehicle.

1. En primer lloc necessitem realitzar l'assignació (marcat a la figura 14 amb un quadrat blau i l'actualització de l'estat amb un cercle del mateix color). Els mòduls usats són:

Search: Busquem de la cua d'accidents una entitat original que compleixi una certa condició. En l'assignació, la condició és que tinguem un accident pendent per ser assignat ($ea=1$) o actualitzat ($ea=4$)⁴. Si en trobem un, l'extraiem de la cua. En cas contrari, eliminem la entitat novament al *dispose*.

Remove: Extraiem de la cua la entitat que volem. Aquest mòdul té dues sortides: la *original* (que és la que ve del mòdul anterior i per tant, és la rèplica que havíem generat) que començarà el procés d'assignació, i la que extraiem de la cua (entitat original) que com ja hem comentat la utilitzarem per actualitzar l'estat de l'accident.

Assign: Actualitzem l'estat de l'accident de pendent a assignat ($ea = 2$) i retornem la entitat a la cua d'accidents.

Process Assignació: Realitzem la assignació amb aquest mòdul junt amb un *delay* que representa el temps transcorregut per a realitzar l'assignació.

2. A continuació entrem a assignar l'accident a un vehicle (quadrat vermell) i la seva corresponent actualització (cercle del mateix color):

Assign: l'atribut *idav* ens guardarà l'identificador de l'accident per assignar-lo a un vehicle⁵.

Search: busquem un vehicle que estigui lliure ($ev=1$) de la cua de vehicles i entrem al següent mòdul. En cas contrari (no l'hem trobat) enviem la entitat al *dispose*.

Remove: si el trobem, a la entitat extreta actualitzem l'estat del vehicle. a ocupat o assignat ($ev=2$) amb un *assign* i la retornem a la cua de vehicles, mentre que la original es mourà fins a la següent línia d'acció.

⁴ Veure línia d'actualització (3.3.7).

⁵ cal recordar que un accident pot tenir assignats molts vehicles, però un vehicle només pot estar relacionat amb un accident alhora (relació 1- n).

Assign: La actualització del vehicle passa de lliure a assignat, és a dir, ocupat ($ev=2$).

Observant la figura veiem un *delay* abans d'entrar a al següent línia d'acció. Aquests *delays* s'usen per a la sincronització⁶.

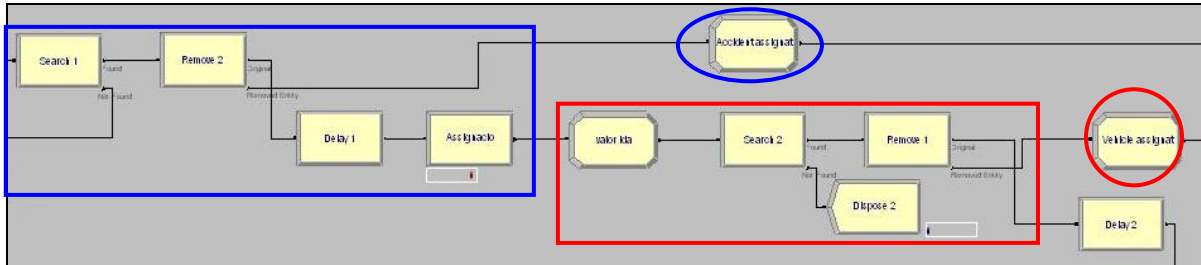


Figura 14: assignació d'un accident i assignació a un vehicle

3.3.6. Línia de moviment dels vehicles

Aquesta línia només fa referència a l'entitat vehicle. Seguirem el mateix procés que anteriorment usant un *search* i un *remove*.

Search: busquem un vehicle a la cua que estigui assignat ($ev=2$) ja que perquè un vehicle entri en moviment aquest ha d'estar assignat. Si no, eliminem la entitat al *dispose*.

Remove: extraiem la entitat original de la cua i actualitzem l'estat del vehicle a en moviment ($ev=3$). L'altra, entra en el *process* Moviment

Process Moviment: simbolitza el temps que el vehicle està en moviment per arribar al lloc de l'accident.

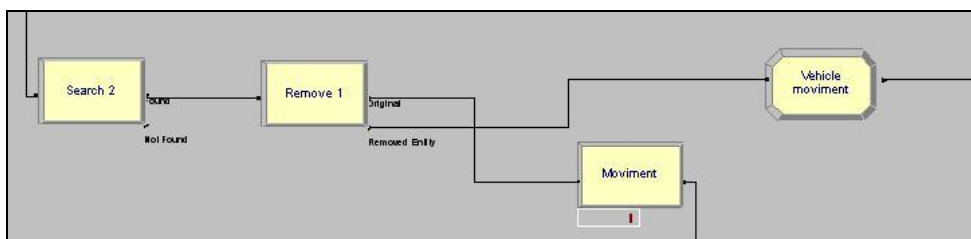


Figura 15: moviment d'un vehicle

⁶ La sincronització és usada en aquelles línies d'acció consecutives que fan referència a una mateixa entitat. Usem un *delay* per assegurar-nos realitzar la actualització de l'estat abans d'entrar al *search* de la línia següent ja que aquest farà la cerca d'aquest valor actualitzat.

3.3.7. Línia d'actualització

Aquesta línia només fa referència a l'entitat accident. Es tracta de l'actualització de les dades una vegada el vehicle ha arribat al lloc de l'accident.

Search: busquem un accident que estigui assignat (ea=2). Si el trobem passem al següent mòdul i sinó enviem la entitat al *dispose*.

Remove: si es compleix la condició extraiem la entitat de la cua d'accidents i realitzem la actualització de les dades entrant al *dispose*. La entitat original (la duplicada inicial) seguirà al procés. Novament tenim un *delay* que ens serveix per sincronitzar el procés.

Decide: amb aquest mòdul decidim amb una condició de percentatge aquells accidents que segueixen el procés o necessiten actualitzar les dades.

Assign Procés: actualitzem l'estat de l'accident a en procés (ea=3).

Assign Actualització: actualitzem l'estat de l'accident a actualització⁷ (ea=4).

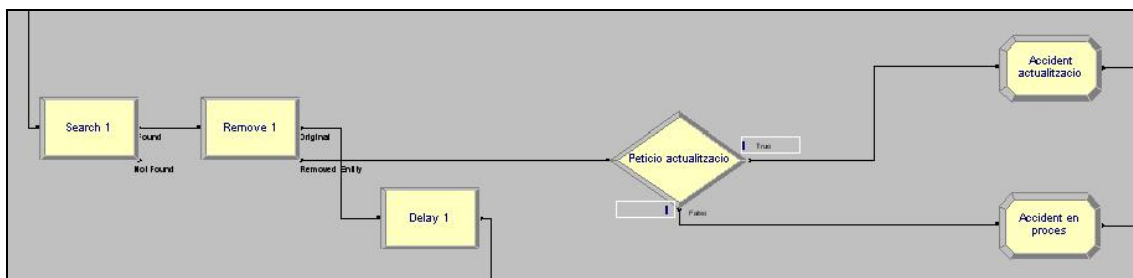


Figura 16: línia de procés o d'actualització

3.3.8. Línia de vehicles ocupats

Aquesta línia fa referència a l'entitat vehicle i només necessitem actualitzar l'estat del vehicle a ocupat (actualment l'estat és en moviment) per a poder començar la intervenció.

Search: busquem un vehicle a la cua que estigui en moviment (ev=3) i entrem al següent mòdul. En cas contrari (no l'hem trobat) enviem l'entitat al *dispose*.

⁷ Aquest estat significa que l'accident necessita actualitzar les dades i ho farem amb una nova línia d'acció que s'explicarà una vegada acabat el disseny del model.

Remove: extraiem la entitat original de la cua i l'enviem a l'actualització. L'altra segueix el seu camí cap a la següent línia d'acció.

Assign: Actualitzem l'estat del vehicle a ocupat (ev=2).

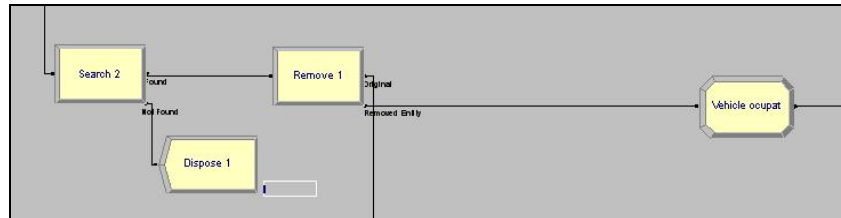


Figura 17: ocupació d'un vehicle

3.3.9. Línia d'intervenció

Una vegada tenim els vehicles ocupats al lloc de l'accident poden començar la tasca d'intervenció. La manera de procedir és la mateixa.

Search: busquem un accident que estigui en procés o actualitzat (ea=3 || ea=4) i entrem al següent mòdul. En cas contrari (no l'hem trobat) enviem la entitat al *dispose*.

Remove: si es compleix la condició extraiem la entitat de la cua i la fem entrar a l'assign.

La entitat original (la duplicada inicial) seguirà al procés.

Decide: amb aquest mòdul decidim amb una condició de percentatge aquells accidents que segueixen el procés o necessiten actualitzar les dades.

Assign Intervenció: actualitzem l'estat de l'accident a en procés (ea=3).

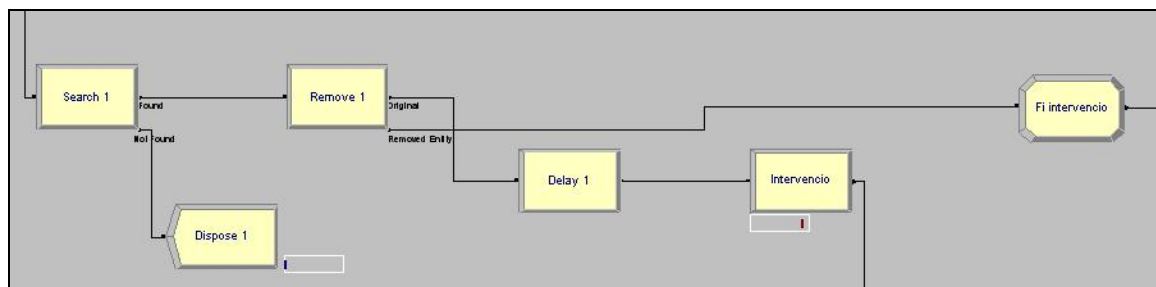


Figura 18: línia d'intervenció

3.3.10. Línia de finalització dels vehicles

Com ja s'ha vist a les XdPC la finalització i retorn cap a la flota és una mica diferent per a cada servei.

Els vehicles de policia poden ser destinats cap a un altre accident sense passar per la flota. Per tant, només necessitarem actualitzar l'estat del vehicle a lliure complint la condició necessària per a poder assignar-lo a un altre accident.

Pel cas dels bombers i les ambulàncies necessitem primer retornar a la flota (vehicle en moviment) i seguidament alliberar el vehicle. A més, les ambulàncies pot ser que enlloc de realitzar el camí cap a la flota necessitin traslladar les víctimes cap a l'hospital si el reconeixement d'aquestes ha resultat greu (ho farem amb un *decide*).

Així doncs per la policia tindrem una línia d'acció, pels bombers dues i per les ambulàncies quatre.

*** Policia:**

Search: busquem un vehicle que estigui ocupat (ev=2). Si no el trobem l'eliminem al *dispose*.

Remove: si el trobem l'extraiem i passa a l'*assign* per actualitzar l'estat del vehicle. L'entitat original la enviem a un *dispose* ja que donarem l'accident per finalitzat.

Assign: Actualitzem l'estat del vehicle a lliure (ev=1)

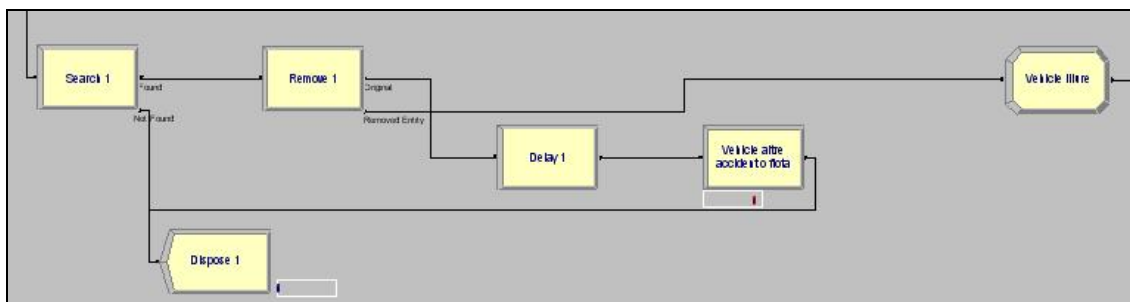


Figura 19: finalització servei dels vehicles de policia

*** Bombers i Ambulàncies:**

Les dues línies comentades seran comunes amb les ambulàncies (moviment cap a la flota i alliberament del vehicle).

Moviment flota:

Search: busquem un vehicle que estigui ocupat (ev=2). Si no el trobem l'eliminem al *dispose*.

Remove: si el trobem l'extraiem i passa a l'*assign* per actualitzar l'estat del vehicle. L'entitat original segueix cap a la següent línia d'acció.

Assign: Actualitzem l'estat del vehicle a en moviment (ev=3).

Alliberament:

Search: busquem un vehicle que estigui en moviment(ev=3). Sinó el trobem l'eliminem al *dispose*.

Remove: si el trobem l'extraiem i passa a l'*assign* per actualitzar l'estat del vehicle. L'entitat original la enviem a un *dispose* ja que ara si que donarem l'accident per finalitzat.

Assign: Actualitzem l'estat del vehicle a lliure (ev=1).

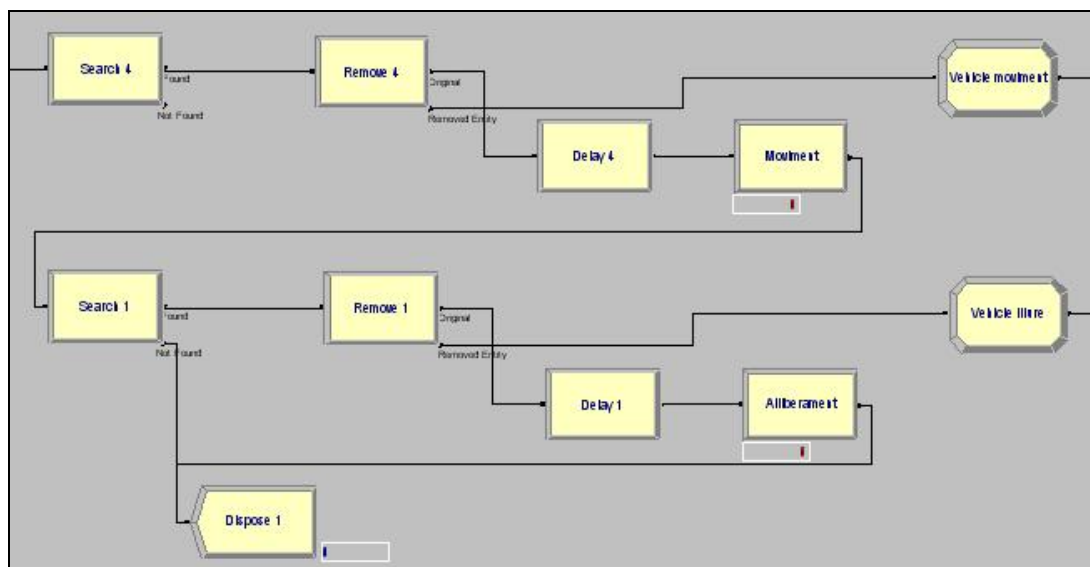


Figura 20: línies comunes finalització servei bombers i ambulàncies (víctimes lleus)

Ambulàncies:

La decisió de si cal traslladar víctimes cap a l'hospital vindrà donada pel valor del color **evi** (víctimes lleus o greus). Aquesta comprovació la realitzarem amb un *decide*.

Decide: La condició d'aquest mòdul serà una expressió: $evi==1$.

Si és **certa** voldrà dir que les víctimes són lleus i per tant donarem el servei per acabat amb les dues línies comunes mostrades a la figura anterior.

Si pel contrari és **falsa** (veure figura 21) voldrà dir que necessitem fer el trasllat cap a l'hospital (estat de les víctimes és greu). En aquest cas haurem de realitzar un moviment però cap a l'hospital i no cap a la flota. Una vegada fet, el vehicle retornarà cap a la flota i serà alliberat.

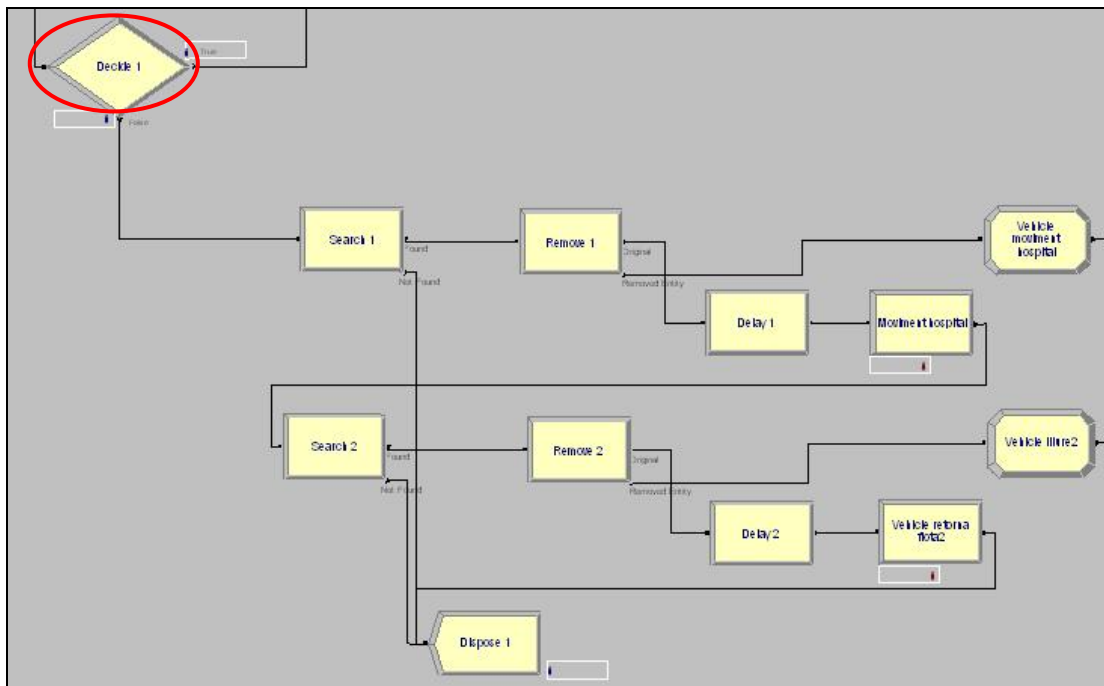


Figura 21: finalització servei en cas d'hospitalització (víctimes greus)

3.3.11. Línia d'actualització de les dades

Una vegada finalitzada l'explicació podem abordar el tema de l'actualització. Com s'ha pogut veure a la línia 3 (línia d'actualització) en el cas que $ea = 4$ (actualitzat) caldrà actualitzar les dades referents a l'accident. Aquesta actualització té lloc just quan hem

arribat al lloc de l'accident, però sabem que les dades també poden canviar en altres moments (sobretot inicialment quan les dades són poc fiables i es realitzen moltes trucades). Per tant, podem haver d'actualitzar les dades quan l'accident estigui encara pendent, assignat o una vegada haguem arribat al lloc de l'accident. Per realitzar l'actualització usarem novament distribucions de probabilitat ja que l'arribada d'informació és aleatòria. Els atributs que caldrà actualitzar⁸ són:

- número de vehicles involucrats (*nvi*)
- número de víctimes atrapades (*nva*)
- número de ferits (*nf*)
- estat de les víctimes (lleus o greus)

Per a poder actualitzar aquests atributs sense alterar l'evolució de l'accident dins el sistema, crearem nous accidents, modificarem els valors d'aquests atributs mantenint l'identificador i l'estat d'aquest, i el retornarem a la cua d'accidents per que continuï el procés.

Els mòduls usats són:

Create: Mateix mòdul que per crear accidents i mateixa funció de probabilitat.

Search: Buscarem aquells accidents que estiguin pendents ($ea=1$) o assignats ($ea=2$) o pendents d'actualització ($ea=4$).

Remove: Exteiem de la cua aquells accidents que compleixin alguna d'aquestes condicions.

Assign: Els atributs modificats usant el *expression builder* es mostren a continuació:

Attribute, evi, ANINT(TRIA(1,1,2))
Attribute, nva, ANINT(NORM(UNIF(1,8),0.5))
Attribute, nvi, ANINT(NORM(UNIF(1,5),0.5))
Attribute, nf, ANINT(NORM(UNIF(1,12),0.5))

⁸ No cal dir que aquests canvis no han de ser canvis dràstics ni situacions impossibles (passar d'un vehicle involucrat a 10 o que totes les víctimes considerades lleus passin a greus). Per fer-ho aplicarem funcions de probabilitat més "suaus" com són la normal o la lognormal (descripció a l'annex B.3).

Duplicate: Amb aquest mòdul tornarem a començar el procés enviant una rèplica a cada servei (*al dispose*).

Finalment enviarem aquests accidents “actualitzats” de nou a la cua d'accidents per a seguir el seu procés.

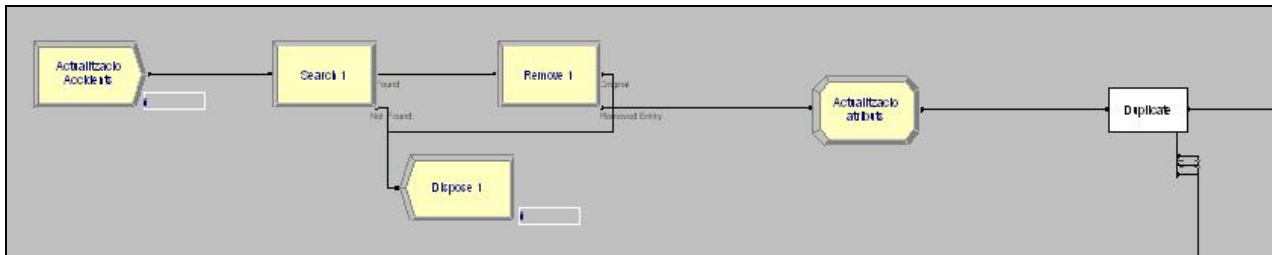


Figura 22: actualitzacions dels accidents

3.4. Representació final del model de Simulació

Tot seguit podem veure una representació del model final per veure les connexions entre les línies d'acció i creació, i les d'aquestes amb les cues d'accidents o de vehicles. Per raons d'espai (el model de representació és molt gran i no podríem distingir els mòduls) només es mostra la branca de Policia.

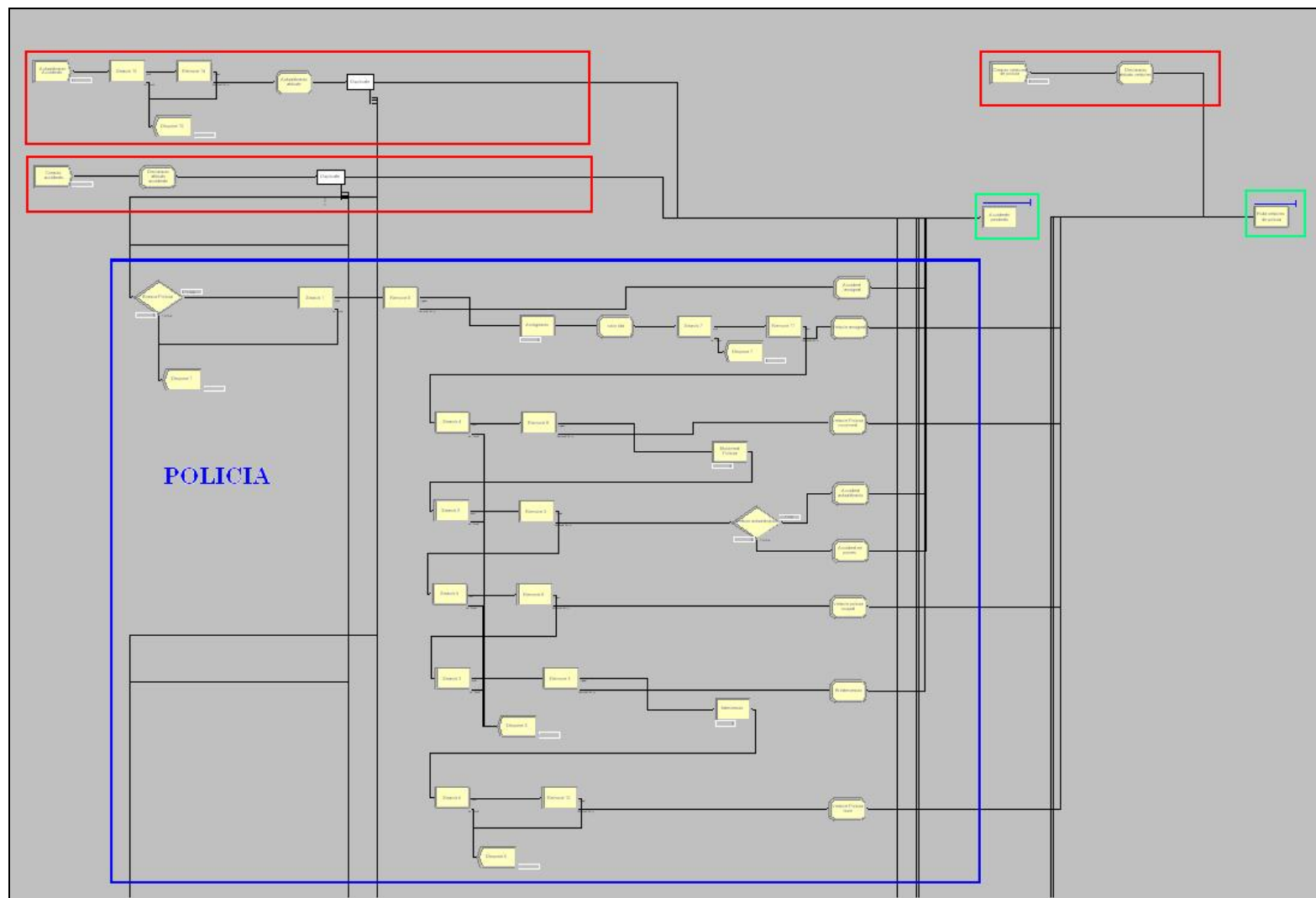


figura 23: Model de Simulació: connexions entre totes les línies de creació i acció amb les cues

4. CONNEXIÓ BDD-CORBA

Abans d'explicar com s'ha realitzat la connexió o en que consisteix la plataforma CORBA, necessitem especificar la base de dades a la qual ens connectarem.

A partir d'Access crearem la nostra base de dades (BDD). La distribució i taules utilitzades es mostren a continuació.

4.1. Especificació de les taules

Les següents taules mostren la informació emprada en la simulació i que necessitarem per atacar la base de dades. Es pot comprovar que la majoria de taules es corresponen als nodes que s'havien definit per a realitzar els models de les XdPA amb tots els seus colors. A continuació es mostra la definició de les taules així com el diagrama amb les relacions entre elles.

1. Accidents

Aquesta taula conté la informació corresponent als accidents que es produeixen en el model de simulació amb Arena. Veiem que els camps corresponen als atributs declarats al node Accident.

Nom	Tipus de dades	Descripció
idAccident	Número	Identificador de l'accident
idNodeAccident	Número	Identificador del node on s'ha produït l'accident
Temps	Decimal	Temps en què es produeix l'accident
TempsMaxim	Decimal	Temps màxim en què cal arribar a l'accident
numVehicles	Número	Nombre de vehicles implicats en l'accident
numVictimes	Número	Nombre de víctimes
numAtrapats	Número	Nombre de víctimes atrapades
EstatAccidentPolicia	Número	Estat de l'accident (1: pendent / 2: assignat / 3: en procés / 4: actualització / 5: finalitzat)
EstatAccidentBombers	Número	Estat de l'accident (1: pendent / 2: assignat / 3: en procés / 4: actualització / 5: finalitzat)

EstatAccidentAmbulàncies	Número	Estat de l'accident (1: pendent / 2: assignat / 3: en procés / 4: actualització / 5: finalitzat)
AssigPolicia	Número	Assignació policia (0: no / 1: sí)
AssigBombers	Número	Assignació bombers (0: no / 1: sí)
AssigAmbulàncies	Número	Assignació ambulàncies (0: no / 1: sí)

2. Víctimes

Aquesta taula conté la informació corresponent a les víctimes dels accidents emprades en la fase d'assignació de recursos.

Nom	Tipus de dades	Descripció
idVictima	Número	Identificador de la víctima
idAccident	Número	Identificador de l'accident on s'ha produït la víctima
idNodeHospital	Número	Identificador del node on es troba l'hospital (0 si no cal trasllat)
EstatVictima	Número	Gravetat de la víctima (1: lleu / 2: greu)

3. Vehicles

Aquesta taula conté la informació corresponent als vehicles disponibles en la flota de cada un dels serveis. Aquesta informació serà utilitzada, principalment, durant la fase d'assignació de recursos.

Nom	Tipus de dades	Descripció
idVehicle	Número	Identificador del vehicle
idServei	Número	Identificador del servei al que pertany el vehicle (1: policia / 2: bombers / 3:ambulàncies)
idNode	Número	Identificador del node on es troba el vehicle
idTipus	Número	Identificador del tipus de vehicle
idAccident	Número	Identificador de l'accident on s'ha assignat el vehicle
idDepot	Número	Identificador de la base pròpia del vehicle
idRuta	Número	Identificador de la ruta assignada al vehicle
EstatVehicle	Número	Estat del vehicle (1: lliure / 2: ocupat / 3:moviment accident o moviment base)

4. Tipus_Vehicle

Aquesta taula conté la informació corresponent als diferents tipus i subtipus de vehicles propis de cada servei.

Nom	Tipus de dades	Descripció
idTipus	Número	Identificador del tipus de vehicle
idServei	Número	Identificador del servei al qual pertany el tipus de vehicle (1: policia / 2: bombers / 3:ambulàncies)
Descripcio	Text	Descripció del tipus de vehicle

5. Serveis

Aquesta taula conté la informació corresponent als diferents serveis presents en el sistema.

Nom	Tipus de dades	Descripció
idServei	Número	Identificador del servei
Descripcio	Text	Descripció del servei / Tipus
Bases	Número	Nombre de bases pròpies de cada servei

6. Bases

Aquesta taula conté la informació corresponent a les bases existents en el sistema que es corresponen amb dades reals de bases de cada servei.

Nom	Tipus de dades	Descripció
idDepot	Número	Identificador de la base
idNode	Número	Identificador del node on es troba situada la base
idServei	Número	Identificador del servei al que pertany la base
Vehicles	Número	Nombre de vehicles ubicats en la base

7. Rutes

Aquesta taula conté la informació corresponent a l'assignació de les rutes calculades pels diferents vehicles, obtinguda a partir de les dades provinents dels clusters definits en l'etapa de preprocés.

Nom	Tipus de dades	Descripció
idRuta	Número	Identificador de la ruta
idVehicle	Número	Identificador del vehicle que segueix la ruta
idAccident	Número	Identificador de l'accident corresponent a la ruta
TempsRuta	Número	Temps total de viatge entre l'origen i el destí de la ruta

8. Taula Rutes_ArcGIS

Aquesta taula conté la informació detallada de cada ruta, amb la seqüència d'arcs GIS que componen la ruta, amb l'objectiu de permetre la simulació del moviment dels vehicles.

Nom	Tipus de dades	Descripció
idRuta	Número	Identificador de la ruta
Sequencia	Número	Posició de l'arc en la seqüència de la ruta
RID	Número	Identificador de l'arc GIS de la secció de ruta

9. Nodos i ArcGIS

Les taules Nodos, ArcGIS corresponen a les taules pròpies de la base de dades estàtica amb informació geogràfica. Aquesta base de dades és comuna a tots els projectes que facin ús del sistema i ja està prèviament definida.

A continuació es mostra l'esquema general de les taules i les relacions entre elles.

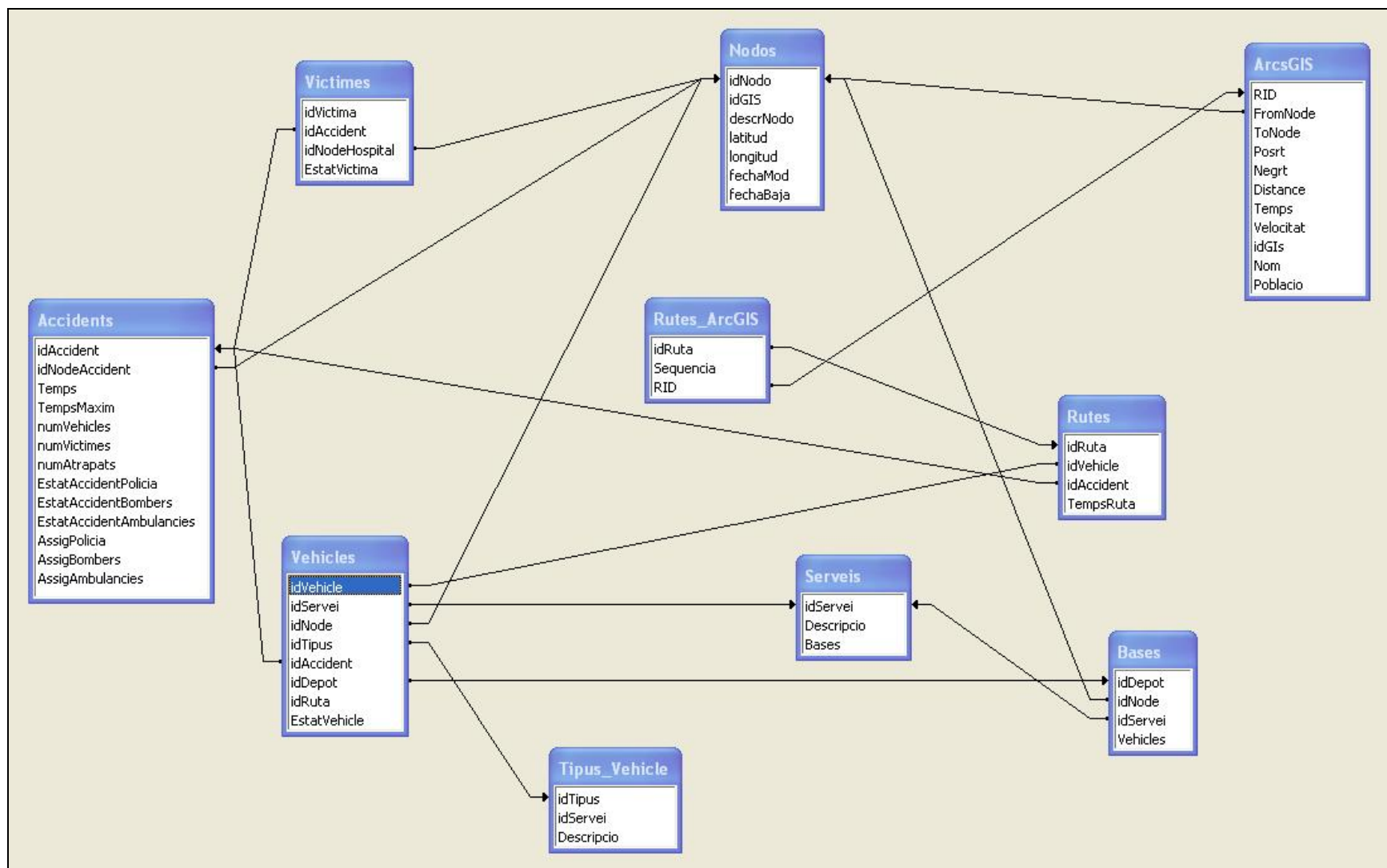


Figura 24: Relacions entre les taules que conformen la Bases de Dades

4.2. Plataformes Distribuïdes

4.2.1. Introducció

El concepte de plataforma distribuïda i concretament el de CORBA tot i semblar molt recent existeix des de fa bastants anys. La seva importància ha vingut directament lligada a partir de la gran difusió que ha experimentat Internet (multitud d'aplicacions, protocols de xarxes que treballen en diferents sistemes operatius, etc...). Aquest creixement ha comportat la necessitat de crear plataformes que permetin a les aplicacions comunicar-se unes amb altres independentment de la seva localització o el seu disseny. La comunicació entre aquestes aplicacions i sistemes tan diferents es fa a partir del protocol TCP/IP.

Els dos paradigmes més coneguts d'objectes distribuïts són **DCOM** (*Distributed Component Object Model*) de Microsoft i **CORBA** (*Common Object Request Broker Architecture*) de la *OMG*.

CORBA és fàcil d'usar en diferents llenguatges com C++ , Java ... i el que s'ha usat és Visual Basic.

4.2.2. CORBA

La arquitectura de CORBA és una tecnologia amb la finalitat de facilitar la creació d'aplicacions distribuïdes amb una interoperabilitat entre components *software* (independentment del llenguatge en que estan desenvolupats), la plataforma *hardware* i el *sistema operatiu* sobre el que s'executen.

CORBA és l'estàndard de la *OMG*⁹ per a la construcció d'objectes distribuïts per qualsevol plataforma, llenguatge o tipus de màquina. Està basada en dos principis:

- La separació de la implementació de la interface del objecte usant **IDL** (*Interface Definition Language*) el qual permet de forma estàndard representar els mètodes i atributs independentment al llenguatge o a la plataforma.
- Protocol **IOP** (*Internet Inter-ORB Protocol*) que permet la invocació sobre objectes remots.

⁹ Defineix una semàntica d'objectes comuna per especificar les característiques visibles externament d'un objecte d'una forma estàndard i independent de la implementació a través d'una interface ben definida especificada en IDL.

Tota la arquitectura CORBA té com a base el **ORB** (*Object Request Broker*) que actua com a bus central d'objectes sobre els quals cada objecte CORBA interactua de forma transparent amb la resta d'objectes CORBA ja siguin locals o remots.

El model d'aplicació **Client/Servidor** (també anomenat de dos capes) és el més usat en l'actualitat. A diferència del monolític (el qual en cada ordinador duplica tots els seus elements), el Client/Servidor separa la interface de l'usuari amb les regles de la gestió de les dades i l'accés a elles. El Servidor de dades (màquina dedicada exclusivament) és l'únic que manipula la base de dades executant les peticions que rep dels clients.

En el nostre cas volem a partir d'una màquina sota Windows XP (client) connectar-nos a un servidor CORBA a partir de l'accés a objectes CORBA mitjançant una DLL. Aquest servidor al qual ens connectarem té com a sistema operatiu la distribució Linux Suse 9.3 i l'aplicació MySQL versió 4.1.10a.

4.2.3. Connexió de components via CORBA mitjançant una DLL

Com ja hem comentat l'avantatge de CORBA és que permet accedir-hi a partir de llenguatges de programació que no disposen de cap distribució de CORBA com és el cas de Visual Basic.

Per accedir a objectes CORBA s'ha utilitzat la llibreria *MySQLCorba.dll* programada en C++ des d'una aplicació en Visual Basic a partir de la creació d'un seguit de mètodes (com per exemple *executeUpdate*). Tot i que no entrarem en detalls¹⁰, cal mencionar que la finalitat és poder accedir a les funcions des de fora d'ella.

4.2.4. Accés a la Base de Dades i invocació a la DLL

La connexió a la base de dades MySQL es realitza gràcies a la possibilitat de realitzar des d'Arena crides a Microsoft VBA (*Visual Basic for Applications*) en forma de blocs **VBA** que s'integren dins del model de simulació com qualsevol altre mòdul.

A partir de la DLL creada s'ha definit la classe *VB CPasarelaCorba*¹¹ que ens permet connectar el codi Visual Basic invocat des del simulador amb les funcions que ofereix la DLL de connexió amb MySQL.

És a dir, podrem utilitzar funcions dins del nostre codi com si fossin funcions declarades en Visual Basic quan en realitat s'utilitzarà la funció que ofereix la DLL.

¹⁰ més informació es pot consultar a l'article *Conexión de componentes via CORBA mediante una DLL* [Carmona, 2006] i afegint la comunicació entre ORBs. *OpenORB i omniORB*

¹¹ Article *Modelo de simulación: Accesos a la base de datos* [Carmona, 2006]

4.2.5. Funcions disponibles

Per a declarar les funcions corresponents amb Visual Basic que ens permetran accedir a la base de dades, farem ús de la següent funció de la DLL

Option Explicit *Sentència que obliga a definir totes les variables*

Private Declare Function **nomVB** Lib "Nom.dll" Alias "NomDLLFunció"
(Paràmetres de la funció) **TipusFunció**

on:

NomVB és el nom de la funció dins de VB.

Nom.dll és el nom del fitxer que conté la DLL (en el nostre cas serà *MySQLCorba.dll*)

NomDLLFunció és l'alias¹² dins de la DLL de la funció cridada

La declaració de les funcions que s'han utilitzat es mostren a continuació:

* **comptador**: Retorna el número de tuples que compleixen la condició sentència.

```
Private Declare Function comptador Lib "MySQLCorba.dll" Alias "_comptador@12"
(ByVal
  host As String, ByVal database As String, ByVal sentència As String) As Long
```

* **executeUpdate**: Funció encarregada d'actualitzar camps de la bases de dades.

```
Private Declare Sub executeUpdate Lib "MySQLCorba.dll" Alias
"_executeUpdate@12"
(ByVal host As String, ByVal database As String, ByVal sentència As String)
```

* **executeQueryLong**: Funció per realitzar la consulta d'un valor de tipus enter.

```
Private Declare Function executeQueryLong Lib "MySQLCorba.dll" Alias
"_executeQueryLong@12" (ByVal host As String, ByVal database As String, ByVal
sentència As String) As Long
```

* **executeQueryDouble**: Funció per realitzar la consulta d'un valor de tipus doble.

```
Private Declare Sub executeQueryDouble Lib "MySQLCorba.dll" Alias
"_executeQueryDouble@16" (ByVal host As String, ByVal database As String,
ByVal
sentència As String, ByVal res As Double)
```

¹² Per conèixer l'alias d'una funció dins de la DLL s'utilitza la comanda DUMPBIN la qual ens permet veure el nom intern de la funció en la DLL i el mètode al que fa referència

* **executeQueryCampsVert**: Funció para realitzar consultes que retornen varies tuples per un únic camp o columna.

```
Private Declare Sub executeQueryCampsVert Lib "MySQLCorba.dll" Alias
  "_executeQueryCampsVert@20" (ByVal host As String, ByVal database As String,
  ByVal
  sentencia As String, ByVal dimensio As String, ByVal res As Double)
```

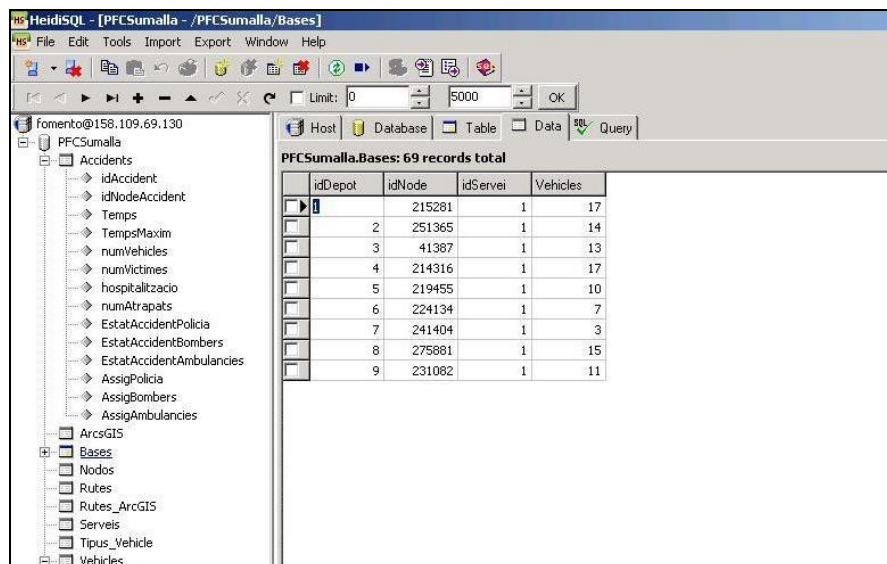
En totes aquestes funcions, els paràmetres host (IP) i database (nom de la BDD) tindran sempre els següents valors: "158.109.69.130" i "PFCSumalla". La sentència serà una línia d'execució que no deixen de ser consultes en SQL.

Per a la visualització de la BDD, hem utilitzat una petita aplicació open-source anomenada **HeidiSQL**¹³ la qual ens permet consultar en tot moment l'estat de la BDD per veure insercions, delacions o modificacions fetes des dels blocs VBA d'Arena.

Un exemple senzill de l'ús d'aquestes funcions seria el següent:

```
comptador("158.109.69.130", "PFCSumalla", "select count(*) from Bases where
idServei=1")
```

On busquem totes aquelles flotes de policia (idServei=1) existents a la taula Bases. En la següent imatge podem veure el resultat d'aquest exemple en l'aplicació.



idDepot	idNode	idServei	Vehicles
	215281	1	17
2	251365	1	14
3	41387	1	13
4	214316	1	17
5	219455	1	10
6	224134	1	7
7	241404	1	3
8	275881	1	15
9	231082	1	11

Figura 25: Interfície gràfica de Heidi: resultat de la consulta

¹³ **HeidiSQL** és un client open source per a MySQL que permet controlar i visualitzar les bades de dades i les taules de MySQL.

5. CONSTRUCCIÓ DEL MODEL VBA

Una vegada definida la connexió i les funcions que usarem per modificar la Base de Dades ja podem centrar-nos en la creació del model VBA en Arena.

Aquest model, simplifica molt el model anterior en nombre de mòduls ja que el pes de la informació dels accidents i vehicles passa a estar a la BDD. L'altre punt important, és la introducció dels blocs VBA els quals ens donen molta més autonomia en el funcionament de la simulació respecte al que volem fer. Part d'aquesta llibertat junt amb la informació que trobem en la BDD (taules predefinides amb informació geogràfica autèntica) ens permet generar una simulació gràfica "real" amb l'ajuda del Google Earth.

5.1. Similituds i canvis respecte al model de Simulació

El model VBA, presentarà el mateix patró a seguir en el transcurs de l'accident. És a dir, la dinàmica de funcionament de les línies d'acció (creació, assignació, moviment...) serà el mateix així com els canvis d'estat tant dels Accidents com dels Vehicles. No cal dir, que també seguirem tenint una branca per cada servei: Policia, Bombers i Ambulàncies.

Respecte els canvis, els més destacats són:

- * **Bloc VBA:** com ja s'ha comentat és un dels canvis principals i és el mòdul que ens permet mitjançant Visual Basic i les funcions abans explicades, modificar el contingut de les taules de la BDD.
- * **Cues d'Accidents i Vehicles (*Hold*):** Aquestes cues, ens servien per emmagatzemar els accidents i vehicles que s'anaven creant així com l'actualització dels estats d'aquests.
En aquest model, tots els canvis dels atributs d'Accidents i Vehicles es fan a les taules respectives de la BDD a partir d'actualitzacions (updates).
- * **Mòduls *Search i Remove*:** Al desaparèixer les cues, ja no necessitem aquests dos mòduls que ens servien per extreure i eliminar l'entitat a modificar de la cua en cada línia d'acció.
- * **entitats Vehicles i creació:** No les farem servir ja que tota la informació relativa als serveis i vehicles està en les taules Bases i Vehicles.
- * **Moviment:** Fins ara el moviment dels vehicles només l'havíem representat amb un *process* que consumia temps. En aquest model, ens centrarem en implementar-lo gràcies a les taules predefinides (Nodos, ArcGIS i Rutes_ArcGIS).

5.2 Línies d'acció i blocs VBA

Les línies d'acció (veure punt anterior) segueixen el mateix format que el primer model de simulació només amb Arena. La única que mereix especial atenció és la línia de Moviment. L'altre punt que ens interessa és saber la funció que realitzen els blocs VBA en cada línia. A continuació, es farà una breu explicació de cada línia d'acció i de tots els blocs VBA¹⁴ que conformen el model definitiu.

Cada cop que una entitat entra en un bloc VBA llencem el codi Visual Basic. L'estructura del codi de tots els blocs consta de:

- Declaració de variables, objectes i funcions internes.
- Funcions predefinides (veure punt 4.2.5) que ens permeten accedir i modificar els valors de les taules de la BDD.
- Declaració en alguns blocs de fitxers .kml¹⁵ per a la representació gràfica.

5.2.1. Creació d'accidents

Una vegada hem creat els accidents i hem definit els atributs ens trobem amb els blocs VBA 1 i 9.

VBA1: En aquest bloc fem la inserció dels atributs de l'accident a la taula Accidents i creem el fitxer Accidents.kml per visualitzar la localització dels Accidents generats a partir de la taula Nodos.

VBA9: Creació dels fitxers (FlotesP.kml, FlotesB.kml i FlotesA.kml) amb la informació geogràfica de les bases dels serveis extreta de les taules Bases i Nodos.

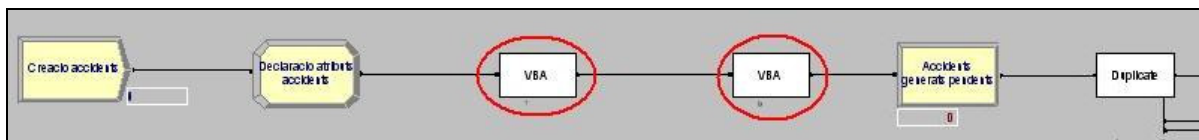


Figura 26: Blocs 1 i 9 de la línia de creació

A la figura 27 es poden veure el resultat de l'execució d'aquests blocs representant els fitxers dels accidents i flotes en Google Earth. Podem veure la localització de dos

¹⁴ En l'annex C.1 trobem una explicació dels blocs VBA i en el C.3 podem trobar la implementació de tots ells (codi Visual Basic de model resultant).

¹⁵ En l'annex C.2 podem trobar una breu explicació dels fitxers kml utilitzats pel Google Earth.

accidents (Accident 1 i Accident 2) i de diverses flotes de Policia, Bombers i Ambulàncies (cadascuna representada pel seu color). La representació s'aconsegueix marcant la posició (latitud i longitud) del node en el qual té lloc l'accident o existeix una flota de la taula Nodos.



Figura 27: Representació geogràfica dels fitxers kml d'accidents i Flotes dels serveis

Tot seguit podem veure un exemple del contingut de la taula Nodos on podem trobar la longitud i la latitud del node que volem representar (en aquest cas el node 45568 correspon a l'Accident2).

Host	Database	Table	Data	SQL Query
PFCSumalla.Nodos: 281.257 records total, 20 matching to filter				
idNodo	idGIS	descNodo	latitud	longitud
45568	37240299426676		2,2214488	41,5545245

Figura 28: Exemple Taula Nodos

5.2.2. Línia d'actualització

En aquesta línia, realitzem l'actualització dels accidents.

VBA17: Sempre que l'estat de l'accident estigui pendent d'assignar, realitzarem una actualització dels atributs que poden sofrir canvis (nvi, nva, nf i evi). Aquestes modificacions es realitzen en la taula Accidents amb l'entitat Accident que està activa en aquell moment.

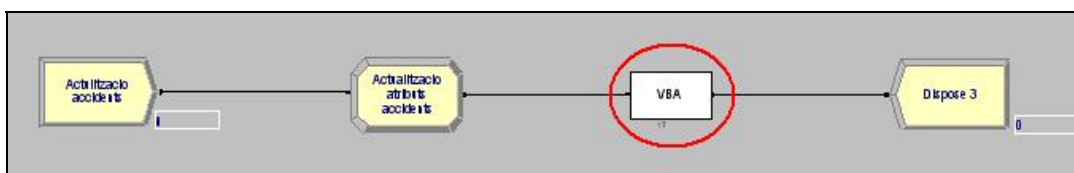


Figura 29: Bloc 17 de la línia d'actualització

5.2.3. Línia d'assignació

Aquesta línia representa els canvis d'estat de l'accident (pendent i assignat). En les tres flotes, el seguiment és el mateix i només actualitzem la taula Accidents.

VBA2,10,18: Com que ja tenim l'accident creat i el duplicate ha enviat una còpia a cada servei, actualitzem el camp EstatAccident(Policia, Bombers o Ambulàncies) a pendent (=1).

VBA3,11,19: Fem una actualització de l'estat de l'accident dels serveis a assignat (=2) i bloquegem l'entrada al servei (AssigPolicia, AssigBombers o AssigAmbulancies).

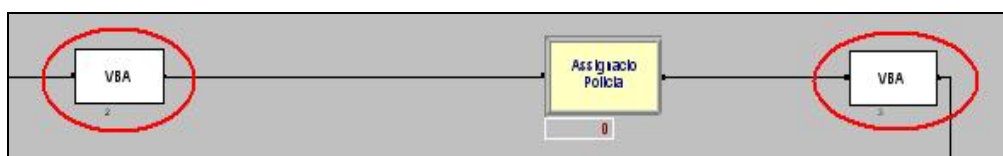


Figura 30: Blocs VBA de la línia d'assignació de Policia

5.2.4. Línia d'assignació i moviment dels vehicles

Aquesta línia realitza dues funcions importants:

1. Assignació de l'accident al vehicle de la flota corresponent i actualització d'estat del vehicle.
2. Realització del moviment cap al lloc de l'accident del vehicle corresponent.

VBA4,12,20: Realitzen la primera funció. Necessitem assignar l'identificador de l'accident en curs al primer vehicle lliure que trobem a la flota del servei. Per fer-ho, agafarem l'identificador de la taula Accidents i l'introduïrem a la taula Vehicles. Actualitzarem l'estat del vehicle a ocupat (=0) de la flota i servei corresponent.

VBA8,13,21: Són els mòduls encarregats d'actualitzar la posició del vehicle a partir dels nodes que anem visitant segons la ruta marcada. A continuació es defineixen els passos a seguir:

- Busquem vehicles del servei assignats que han de començar el moviment (relació entre les taules Accident i Vehicles).
- Consultem els valors dels nodes als quals hem d'anar accedint segons quin sigui l'accident (relació entre les taules ArcGIS, Rutes_ArcGIS i Vehicles).
- Actualitzem la posició del vehicle en cada moviment segons si el node al que volem anar coincideix amb el toNode o el fromNode.
- Creació del fitxer de moviment dels vehicles de cada servei: MovimentP.kml, MovimentB.kml i MovimentA.kml. El contingut d'aquest fitxer són les coordenades del node en el qual es troba el vehicle i s'anirà refrescant per poder visualitzar el moviment "real" en GoogleEarth.

Cal remarcar la importància del *decide* (indicat en blau en la figura 31) el qual té com a condició de sortida (=1) quan el node actual en el que ens trobem sigui el mateix que el que s'ha produït l'accident. Mentre no haguem arribat al lloc de l'accident, seguirem a mode de bucle visitant els nodes que ens marca la ruta triada.

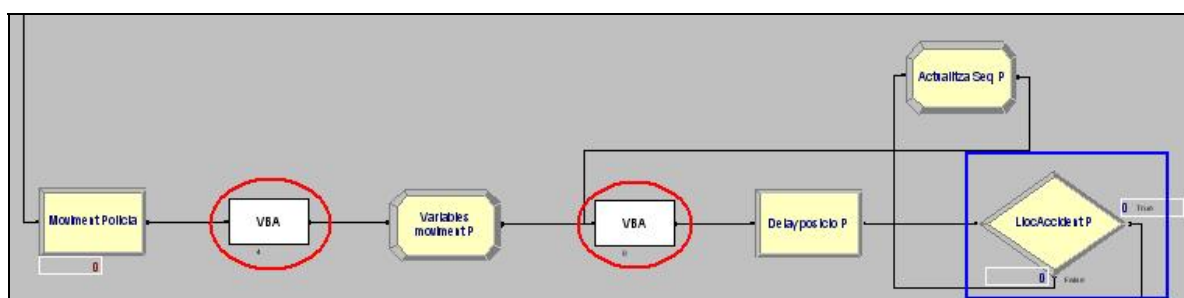


Figura 31: Línia d'assignació i moviment dels vehicles de la flota de Policia

A la figura 32, es poden veure dos vehicles (Policia i Ambulàncies) realitzant el moviment cap al lloc de l'accident (Accident 2). El contingut dels fitxers kml MovimentP i MovimentB es van refrescant a partir del fitxer *GeneraMov* (comentat en l'annex C.2.2).



Figura 32: Exemple del moviment dels vehicles des de GoogleEarth

5.2.5. Línia d'intervenció

Línia en la qual el servei d'emergència comença la seva operació.

***Policia i Bombers:**

VBA5,14: Modifiquem l'estat de l'accident del servei a ocupat (=3) en la taula d'Accidents.

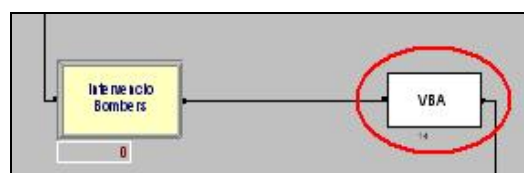


Figura 33: Blocs VBA de la línia d'intervenció de bombers

***Ambulàncies:**

En el cas del servei d'ambulàncies, com que realitzem un primer reconeixement i existeix la possibilitat que la víctima necessiti hospitalització, en la mateixa línia, comença i acaba la intervenció del servei. Per tant, el bloc **VBA22** realitza la mateixa funció que els anteriors i només canvia l'estructura de la línia en Arena.

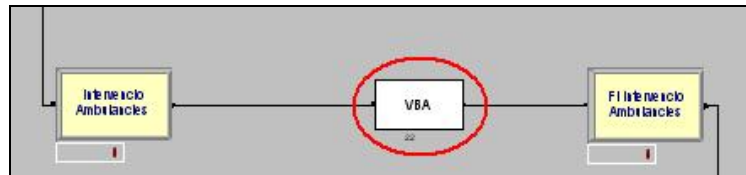


Figura 34: Línia d'intervenció de les ambulàncies

5.2.6. Línia de finalització del servei

***Policia i Bombers:**

Una vegada realitzada la intervenció, el servei és dóna per acabat en el cas de Policia i Bombers. La diferència és que en els vehicles de Policia poden ser assignats a una altre accident sense passar per la flota en cas de ser necessari. L

VBA6,15: Donem la intervenció per acabada. Actualitzem l'estat a finalitzat (=5) en la taula Accidents.

VBA7,16: Actualitzem les dades dels vehicles per retornar-los al seu estat inicial. Camps idaccident=0, estat=1 (lliure) de la taula Vehicles.



Figura 35: Línia de finalització del servei de Policia

***Ambulàncies:**

Donat que en la línia anterior ja havíem donat el servei per acabat, seguint el model de simulació en Arena, entrarem en el *Decide* per si cal hospitalitzar les víctimes ($evi=1$). Si no és necessari, de la mateixa manera que els altres serveis, enviarem l'ambulància cap a la base (blocs 23 i 24). En cas contrari, realitzarem el moviment cap a l'hospital i

una vegada fet, retornarem el vehicle a la flota (blocs 25 i 26). En la següent figura, podem veure l'estructura general.

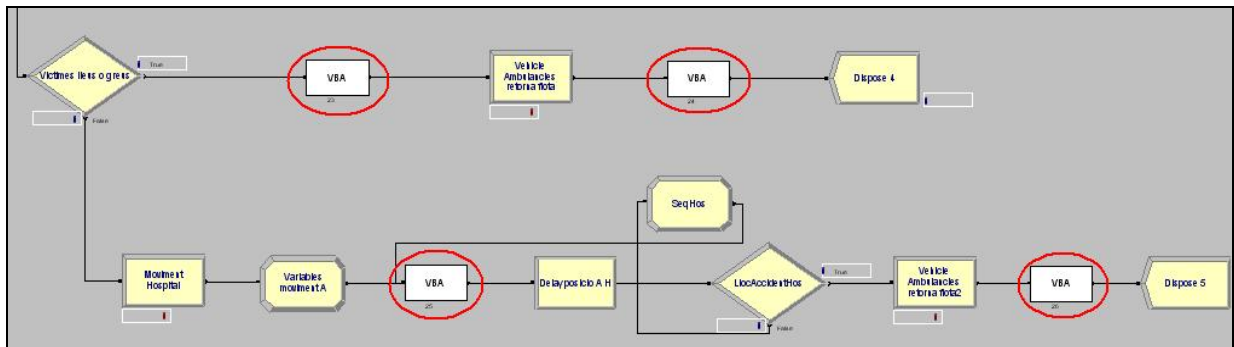


Figura 36: Línia de finalització del servei d'Ambulàncies amb hospitalització

5.3 Representació final del model VBA

Una vegada explicada cada línia d'acció i els respectius blocs, ja podem representar el model definitiu.

A continuació es pot veure el model final (figura 37).

En el CD podem trobar el fitxer amb el model resultant (SimulacioFINALVBA.doe). També s'ha inclòs una gravació feta del GoogleEarth a partir de l'execució de la simulació en la qual podem veure la generació de dos accidents i l'enviament de vehicles des de les flotes dels serveis.

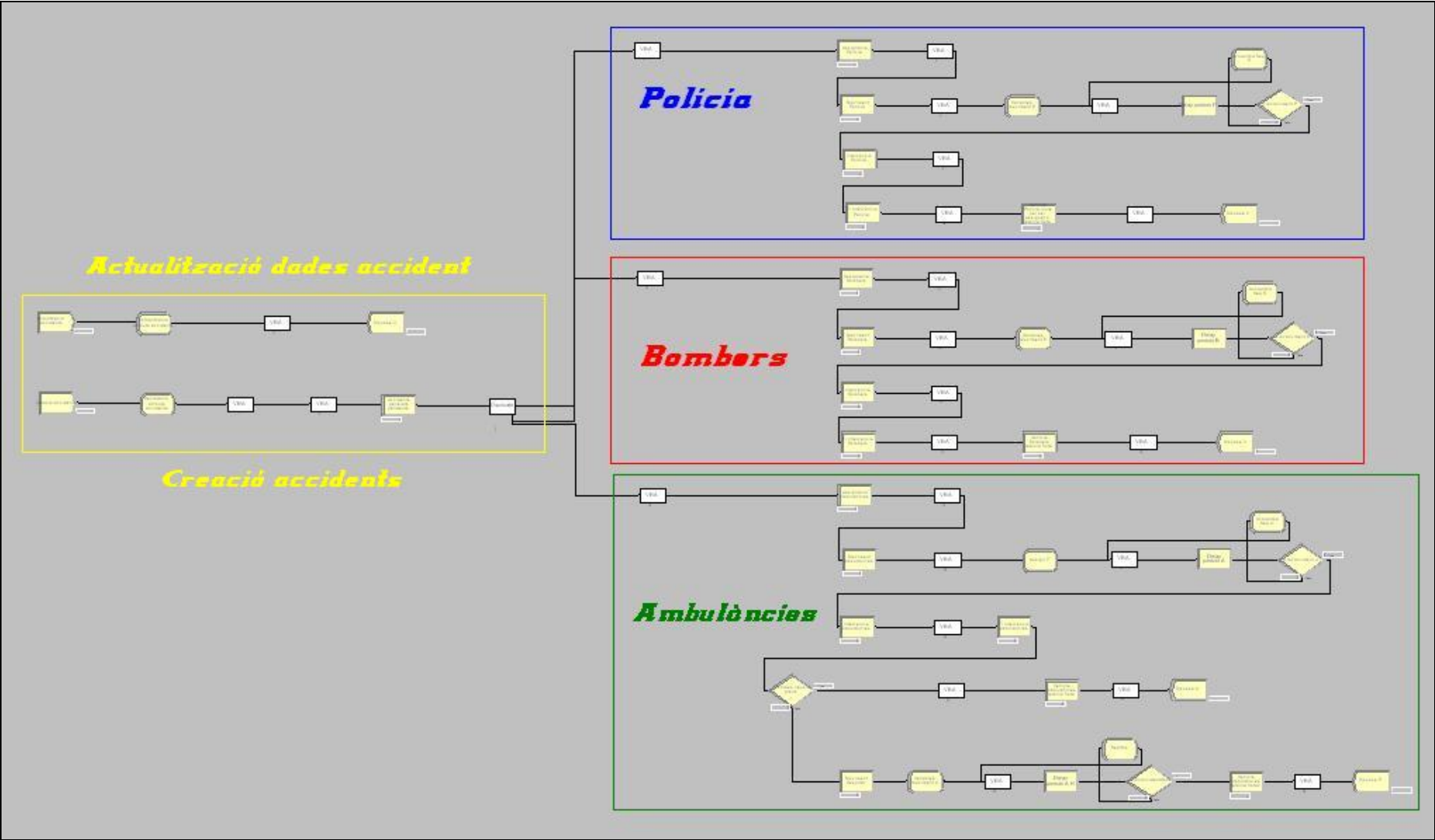


Figura 37: Esquema general del model VBA

6. CONCLUSIONS

Les Xarxes de Petri Acolorides són una eina de modelat de gran flexibilitat. Serveixen perfectament per modelar un sistema a priori tan complex com el descrit en aquest projecte.

Els resultats aconseguits en els models de representació són molt complets i assoleixen tots els punts que s'havien marcat com a objectius.

El software Rockwell Arena 7.0 és un potent programa de simulació. La seva facilitat d'ús, flexibilitat i capacitat de modelatge han permès realitzar els models de simulació que es volien aconseguir (especialment el segon) junt amb l'ús de les funcions de distribució per generar i actualitzar accidents aleatòriament.

Tot i així, aconseguir la completesa que ens ofereixen les Xarxes de Petri és força més complicat.

El primer model fet únicament amb Arena ha presentat certes dificultats en el seu desenvolupament ja que ha costat interpretar els resultats de la simulació únicament des d'Arena.

Els dos objectius principals eren aconseguir l'assignació i la coordinació. El primer s'ha complert satisfactòriament mentre que el segon ha estat més complicat en part, per la obligatorietat de retornar i extreure les entitats (accidents i vehicles) de les cues. Aquesta dificultat s'ha resolt amb escreix al suprimir les cues per les taules de la base de dades.

El segon model (simulació VBA) culmina l'aprenentatge del programa amb la realització d'un model més complet i funcional gràcies a dos fets fonamentals: la incorporació del llenguatge Visual Basic i la connexió via Corba a la base de dades.

La combinació del llenguatge Siman d'Arena amb Visual Basic ens permet treballar en dues direccions: exportar el contingut de les entitats a les taules de la base de dades o realitzar consultes d'aquestes per importar-les a la simulació.

Com a resultat podem aprofitar la informació geogràfica (nodes i coordenades) de les taules per representar el conjunt de flotes i accidents i el moviment dels vehicles.

Aquesta representació s'aconsegueix gràcies a la creació de fitxers kml de Google Earth.

L'exemple de simulació reflecteix l'èxit en la generació del moviment a partir d'accidents i rutes predefinides.

Un aspecte a millorar en aquest segon model seria re calcular el nombre de víctimes per a l'enviament de reforços al realitzar l'actualització de les dades.

Un altre punt seria la millora del contingut de les taules de la base de dades. Per una banda, afegir noves rutes per poder veure conjuntament els tres serveis en acció davant d'un accident i per l'altre, afegir-ne pel trasllat de les víctimes a l'hospital.

La utilitat del projecte dins del projecte del departament assoleix els objectius marcats tot i que encara no s'hagi integrat del tot.

BIBLIOGRAFIA

Carmona, C., *Conexión de componentes vía CORBA mediante una DLL*, LogiSim, 2006.

Carmona, C., *Modelo de simulación: Accesos a la base de datos*, LogiSim, 2006.

Clarke, J., *Simulation Modeling & Analysis with Arena*, 1st edition, Laurier, 2005.

Dicesare, F. [et al.], *Practice of Petri Nets in manufacturing*, Chapman & Hall, 1993.

Guasch, A., [et al.], *Modelado y Simulación: Aplicación a procesos logísticos de fabricación y servicios*, Segunda Edición, Edicions Upc, 2003.

Guasch, T., Puig, V., Figueras, J., *Simulación de procesos logísticos y retos para la optimización de flotas en tiempo real*, Logisim, 2006.

Guimarans, D., Ramos, J.J., Piera, M.A., Guasch, A., *A simulation based decision tool to coordinate emergency services in a road accident*.

Kelton, W. David [et al.], *Simulation with Arena*, 2nd edition, McGraw-Hill, 1998.

Puig, V., Guasch, A., Piera, M.A., Figueras, J., *Routing Vehicles Problems, Algorithms and Systems: A State of the Art (Actividad 5: Routing vehicles problems)*, LogiSim, 2003.

Ramos, J.J., Guimarans, D., Piera, M.A., Guasch, A., *A technological platform for designing real-time decision tools in transportation logistics*.

Fonts electròniques:

Arena: <http://www.arenasimulation.com/idealbb/search.asp>

Visual Basic: <http://www.visual-basic.com.ar/links-es.htm>

MySQL: <http://www.java2s.com/Tutorial/MySQL/CatalogMySQL.htm>

Google Earth: fixters kml <http://earth.google.es/kml/>

ANNEXES

ANNEX A. Sistemes i Models: descripció i classificació

A.1. Sistemes

Entenem per *sistema* com aquella col·lecció d'objectes o entitats que interactuen entre elles per assolir un objectiu concret.

Estat d'un sistema: Conjunt mínim de variables necessàries per caracteritzar o descriure tots aquells aspectes d'interès del sistema en un cert instant de temps. Aquest conjunt de variables formen les *variables d'estat*.

Els sistemes els podem classificar en:

- **Continus:** Les variables d'estat del sistema evolucionen d'una manera continua al llarg del temps. En són exemples el sistema de control de temperatura d'una habitació o el control del nivell d'un dipòsit.
- **Discrets:** Les propietats d'interès del sistema canvien en certs instants de temps i perduren constants la resta de temps. Aquestes seqüències d'instants on es pot produir un canvi en l'estat del sistema acostumen a seguir un patró periòdic.
- **Sistemes orientats a events discrets:** Segueixen les mateixes propietats que l'anterior (canvis en certs instants de temps) però les seqüències d'instants obeeixen a patrons aleatoris.
- **Sistemes combinats:** Aquells sistemes que combinen sistemes continus o discrets segons interressi. S'acostumen a utilitzar en aquells sistemes els quals tenen parts (a diferència de la resta de components) que obligatòriament s'han de modelar amb algun dels dos tipus.

A.2. Models

La descripció del conjunt de característiques d'interès del sistema que havíem definit en l'estat del sistema configuren el *model del sistema*. El procés d'abstracció per obtenir aquesta descripció es coneix com a *modelat*.

Entenem per *modelatge* o *modelat* el procés d'abstracció per obtenir aquesta descripció.

Existeixen diferents tipus de models (físics, mentals o simbòlics) per representar els sistemes.

Els models físics poden ser estàtics o dinàmics mentre que els mentals s'usen per a tasques de planificació i presa de decisions.

Els models simbòlics poden ser de dos tipus: no matemàtics (lingüístics, gràfics, esquemàtics) o matemàtics (els que utilitzarem).

Els models simbòlics matemàtics mapegen les relacions existents entre les propietats físiques del sistema que es pretén modelar en les corresponents estructures matemàtiques. Normalment els representem en un ordinador. El seu anàlisi el podem dividir en dos tipus:

- models **simples**: eines matemàtiques tradicionals.
- models **complexes**: no existeixen solucions matemàtiques exactes. Fem ús de la simulació.

Factors a tenir en compte per realitzar un bon model:

- Un bon model ha de representar adequadament aquelles característiques del sistema que són del nostre interès.
- Ha de ser una representació abstracta de la realitat senzilla per a facilitar el seu manteniment, adaptació i reutilització.

A.2.1. Classificació dels models

1. Respecte el temps:

- **Estàtics:** S'utilitzen per representar el sistema en un cert instant de temps.
- **Dinàmics:** Permet deduir com les variables d'interès del sistema en estudi evolucionen amb el temps.

2. Respecte a les seves sortides:

- **Deterministes:** Aquells models on tenim un únic conjunt de valors de sortida per un conjunt d'entrades conegudes. Un nou estat pot ser completament definit a partir de l'estat previ i les seves entrades.
- **Estocàstics:** El model no genera un únic conjunt de sortides quan s'utilitza per realitzar un experiment sinó que els resultats s'utilitzen per estimar el comportament real del sistema.

3. Respecte la representació de les variables d'interès:

- **Continus:** Representen l'evolució de les variables d'interès de forma continua.
- **Discrets:** Representen l'evolució de les variables d'interès de forma discreta.

Els models d'events discrets són models dinàmics, estocàstics i discrets en els que les variables d'estat combinen el valor ens instants no periòdics de temps. Aquests instants de temps es corresponen amb la ocurrència d'un event. Entenem per *event* com una acció instantània que pot canviar l'estat d'un model.

ANNEX B. Estructura del programa Rockwell Arena

B.1. Disposició de les finestres

Com molts altres programes i sistemes operatius, Rockwell Arena versió 7.0 basa la seva estructura en funció de diferents finestres i menús desplegable.

Quan obrim el programa, ens trobem amb 3 finestres diferents: **project bar** (on trobem les diferents eines per a crear un model), **split screen** (finestra on realitzem el model) i sota aquesta, **spreadsheet view** (una finestra opcional que conté la informació del model i permet l'accés a tots els paràmetres i elements del model). Se la considera opcional ja que els camps d'un bloc també es poden veure clicant sobre aquest en la split screen).

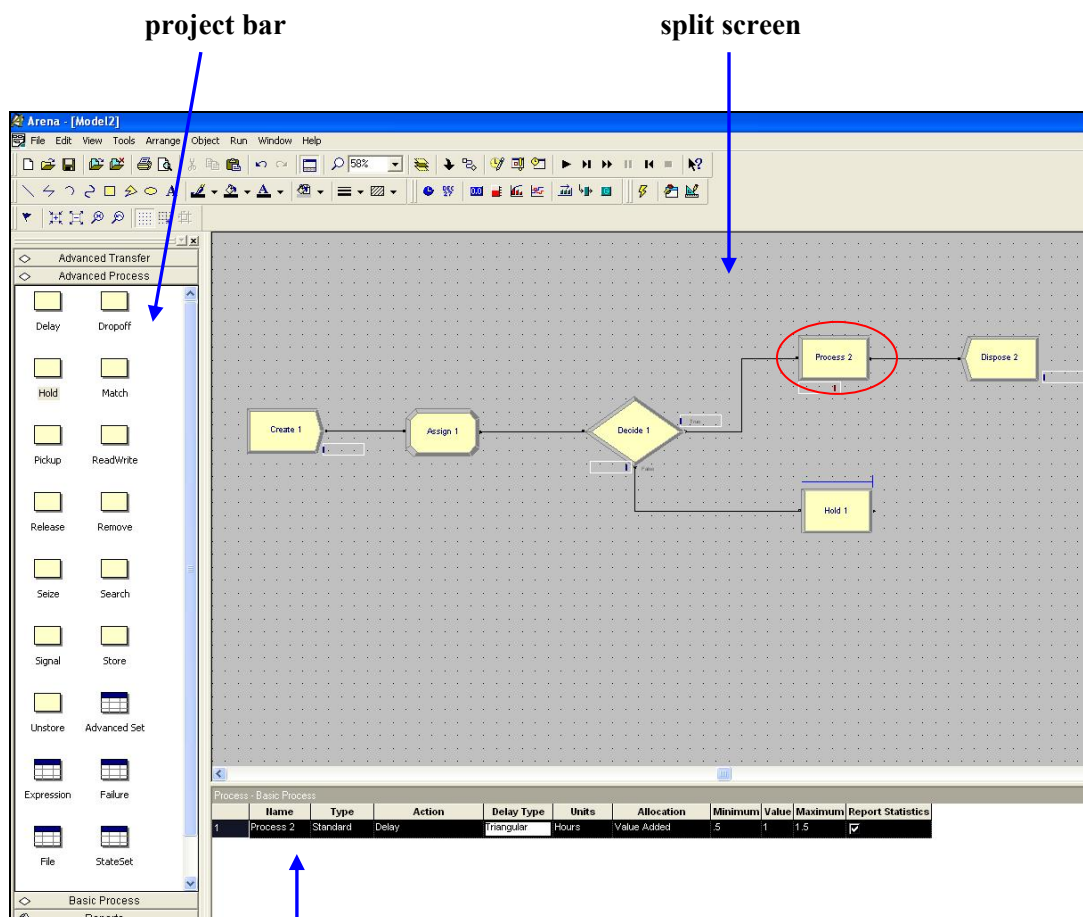


Figura 38: Diferents finestres del programa

spreadsheet view: Visualització dels camps del bloc process

B.2. Estructura dels mòduls

Arena divideix la finestra *project bar* (barra de projectes) en cinc panells:

- **Basic Process Panel:** trobem els blocs bàsics per a construir els models.
- **Advanced Process Panel:** Trobem blocs més complexes i concrets.
- **Advanced Transfer Panel:** Trobem aquells blocs específics per a implementar transports.
- **Blocks Panel (SIMAN template):** Proporciona un accés complet al llenguatge de simulació SIMAN.
- **Reports Panel:** Conté una llista dels resultats de la simulació una vegada acabada la simulació.
- **Navigate Panel:** Permet mostrar diferents vistes d'un model, inclòs diferents submodels en un moment jeràrquic.

Dins dels tres primers panells trobem dos tipus diferents de mòduls:

- **Flowchar Module:** aquells blocs que usem per a construir el model “arrossegant-los” a la split screen i es connecten entre si conformant la lògica del model. (Corresponen a la primera línia de mòduls de la figura 39).
- **Data Module:** blocs que no apareixen a la split screen sinó a la finestra opcional (spreadsheet view) i podem editar els seus camps en unes taules tipus fulls de càlcul. (Corresponen a la segona línia de mòduls de la figura 39).

Els blocs o mòduls descrits a continuació constitueixen els elements bàsics per a construir qualsevol tipus de model de simulació i concretament el realitzat. És necessari tenir un coneixement d'aquests per a poder entendre com s'ha fet. De tots els existents (veure figures) s'expliquen aquells que s'han utilitzat en alguna part del model¹⁶.

¹⁶ Part d'aquesta descripció s'ha realitzat a partir de l'ajuda que ofereix Arena la qual és força extensa i ben estructurada, però a vegades insuficient (i en anglès). Arena té dos tipus d'ajuda: *Arena help online* (on trobem descripcions, relacions, exemples..) i una de més completa en forma de manuals (*Arena Basic Edition User's Guide, Standard Edition i Professional Edition Reference Guide*) segons el nivell de l'usuari.

B.2.1. Basic Process Panel

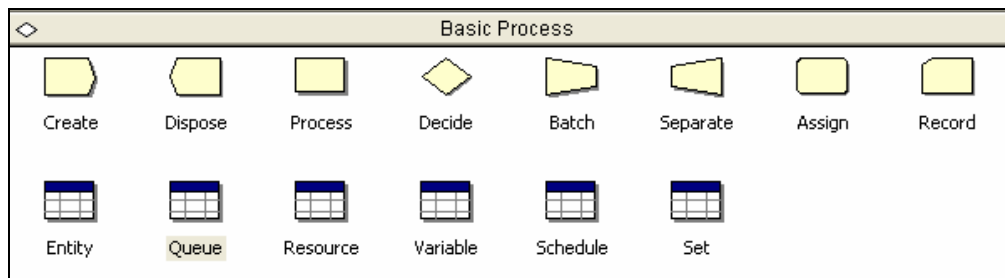


Figura 39 : blocs del panell Basic Process

Create: Representa l'arribada de les entitats al model de simulació.

Dispose: Representa la sortida de les entitats del sistema. És el punt de finalització per les entitats en el model de simulació.

Process: Representa el mètode de processament en la simulació. S'utilitza per a definir una acció que consumeix temps. Pot demanar i/o alliberar un determinat recurs que sigui necessari per a realitzar l'acció.

Decide: Permet prendre decisions en el sistema a partir de condicions definides per l'usuari o a partir de probabilitats.

Batch: S'utilitza per unir un nombre d'entitats ja sigui permanentment o de forma temporal.

Separate: Permet fer còpies d'una entitat ja existent en el sistema. També pot separar un conjunt d'entitats prèviament unides per un bloc *Batch*.

Assign: Permet assignar valors a les variables, atributs d'entitats, tipus d'entitats, icones que representen entitats o qualsevol altra variable del sistema. En un mateix mòdul podem realitzar múltiples assignacions de diferents tipus.

Record: Permet generar estadístiques en el model de simulació ja sigui sobre les entitats o sobre mòduls (processos, cues, create,...).

Entity: Defineix els tipus d'entitats.

Queue: Defineix el tipus de cua establint certes propietats com pot ser la política amb la que entren o surten les entitats, si té o no límit de capacitat, etc.

Resource: Defineix els recursos del model, la seva capacitat i el seu estat inicial.

Variable: Defineix les diferents variables del sistema, la seva dimensió i el seu valor inicial.

B.2.2. Advanced Process Panel

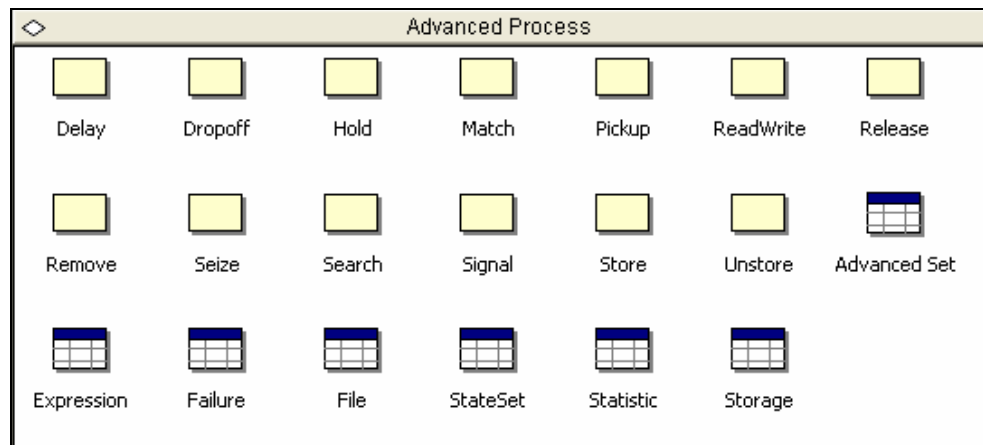


Figura 40 : blocs del panell Advanced Process

Delay: Mòdul que provoca un retard en el transcurs de les entitats pel sistema. S'acostuma a usar per temes de sincronització. Quan una entitat arriba a aquest mòdul s'hi ha d'estar tan temps com s'hagi predefinit.

Hold: Representa el diferents cues d'espera (en el nostre representaran accidents pendents i flotes de vehicles).

Remove: Permet extreure una entitat des d'una posició específica de la cua i la envia al següent mòdul que indica el model.

Search: Permet buscar en una cua una entitat que satisfaci una condició de cerca especificada. Si l'aconsegueix retorna la posició en la cua de la entitat localitzada, en una variable global anomenada **J**.

Signal: Serveix per enviar senyals d'activament a les entitats d'una cua (Hold).

Expression: Permet definir expressions i els seus valors associats. Les expressions poden utilitzar-se en el model mitjançant el seu nom i opcionalment poden ser especificades com arrays unidimensionals o bidimensionals.

StateSet: Defineix els estats dels diferents recursos del sistema.

Statistic: Per generar estadístiques addicionals que volen obtenir-se del model.

B.2.3. Advanced Transfer Panel

D'aquest panell tot i que finalment no surt cap mòdul en el model final, es van realitzar proves amb aquests dos:

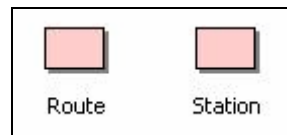


Figura 41 : blocs del panell Advanced Transfer Panel

Route: Permet transferir una entitat a una estació específica del model.

Station: Defineix una o un conjunt d'estacions corresponents a una localització física o lògica on té lloc el procés.

B.2.4. Blocks Panel (SIMAN template)

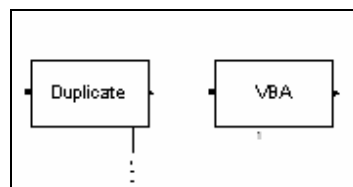


Figura 42 : blocs del panell Blocks Panel

Duplicate: Crea duplicats de les entitats que li arriben i les envia al següent mòdul o a un d'específic.

VBA: Aquest mòdul permet a una entitat disparar un event VBA durant la simulació. En la primera simulació no l'utilitzarem, però si en la segona (simulació VBA).

B.3. Funcions de distribució de probabilitats

Per a realitzar l'assignació d'atributs a les entitats així com en l'actualització de les dades de l'accident fem servir funcions de distribució de probabilitats. L'ús d'aquestes funcions que generen valors aleatoris¹⁷ són el que converteixen el sistema en independent i estocàstic. Una error seria actualitzar les dades fent ús de rèpliques de les entitats i modificant els valors dels seus atributs ja que així aconseguiríem un model massa determinista.

Una **distribució de probabilitats** és un **model teòric** que descriu la forma en que varien els resultats d'un experiment aleatori. Permet relacionar un conjunt de valors o mides amb una freqüència relativa d'aparició. No s'ha de confondre amb:

- **funció de probabilitat** que és aquella funció que assigna probabilitats a cadascun dels valors d'una variable aleatòria discreta.
- **Funció de distribució** que acumula probabilitats associades a una variable aleatòria.

Arena ens permet crear totes les distribucions de probabilitat fàcilment sense haver d'escriure fórmules gràcies al **Expression Builder**. Aquest constructor d'expressions és una petita aplicació on podem seleccionar qualsevol distribució de probabilitats (*Random Distributions*) i usar-la omplint els camps predefinitos que corresponen als paràmetres. A partir d'aquí podem definir combinacions de distribucions o aplica'ls-hi funcions matemàtiques (arrodoniments, valors absoluts, etc.). En la figura següent es pot veure l'aplicació:

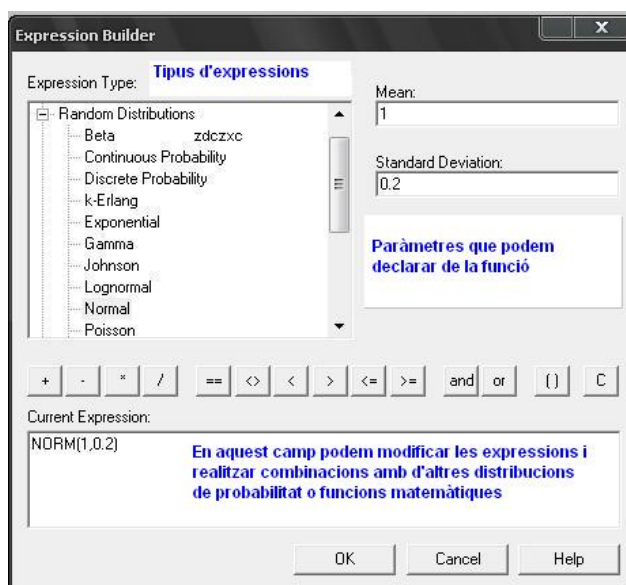


figura 43: Expression Builder

¹⁷ Una variable és aleatòria si pot prendre qualsevol valor no previsible d'un rang finit (variable aleatòria discreta) o infinit (variable aleatòria contínua) de possibles valors.

En l'ajuda d'Arena tenim unes petites nocions de com construir aquestes distribucions de probabilitat. Una informació més detallada, exemples i comparacions d'aquestes funcions les podem trobar a *Simulation Modeling & Analysis with Arena* [J.Clarke, Laurier, 2005].

Un breu resum (sense entrar massa en detalls) de les que s'han utilitzat i perquè, són:

B.3.1. Distribució Normal

També coneguda com a campana de Gauss. La usarem per a realitzar les actualitzacions dels atributs dels accidents. Aquest tipus de distribució permet definir un valor amb una mitjana m i aplicar-li una desviació estàndard.

Notació Arena: **NORMAL(Mean,SD[,Stream])** on *Mean* és la mitjana, *SD* Desviació estàndard i *Stream* és un camp opcional (present en totes les formules i que no utilitzarem) per definir conjunts de valors continguts entre els dos primers.

Aplicant com a mitjana el valor de l'atribut i una desviació d'una unitat, aconseguirem valors actualitzats no molt diferents dels originals que és el que ens interessa.

Un exemple de declaració usada és el següent: $ANINT(NORM(nva,1))$

En la següent figura podem veure una representació de la distribució normal estàndard amb funció de densitat 1.

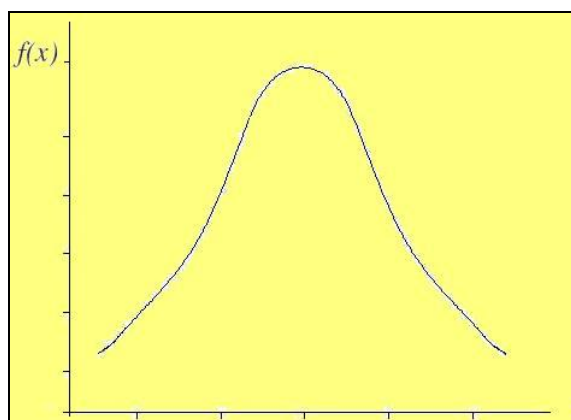


Figura 44 : funció de distribució normal estàndard

B.3.2. Distribució Triangular

És usada quan desconeixem la forma de la distribució però si l'estimació d'uns valors. Aquests valors són tres (mínim, més freqüent i màxim) i ens han servit per definir el tipus de vehicle a seleccionar (policia 1, bombers 2, ambulàncies 3).

Notació Arena: TRIAngular(Min,Mode,Max[,Stream]) i els valors que s'han usat són:

Mínim: 1

Màxim: 3

Més freqüent (o mitjana): 1 (ja que el servei de policia està present en tots els accidents)

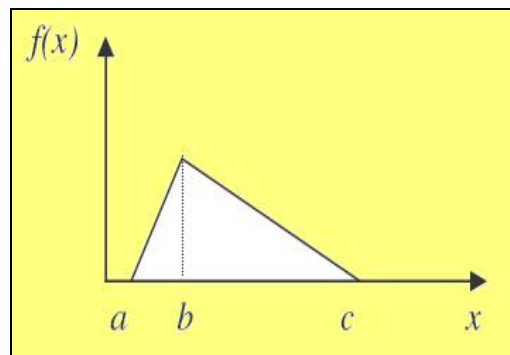


figura 45: exemple de distribució triangular on a és el mínim, b el més probable i c el màxim

B.3.3. Distribució Uniforme

Aquesta distribució s'utilitza quan dins d'un rang conegut tots els valors tenen la mateixa possibilitat d'ocórrer. Els paràmetres a definir són un valor màxim i un valor mínim.

Notació Arena: UNIFORM(Min,Max[,Stream])

Clars exemples per al seu ús són la generació de víctimes atrapades, número de vehicles involucrats i número de ferits on establim un màxim i un mínim però tots el valors tenen la mateixa probabilitat.

Usant aquesta distribució, però, generarem valors amb decimals no desitjats ja que volem nombres enters de víctimes o de vehicles. Per a solucionar-ho combinarem aquestes distribucions amb la funció matemàtica ANINT que arrodoneix el valor a l'enter més proper.

Cal mencionar que la distribució uniforme entre 0 i 1 té una aplicació molt important en simulació ja que permet obtenir valors d'una distribució qualsevol.

La manera d'aconseguir-ho és a partir de la funció de distribució acumulada de la distribució que volem simular i es construeix la seva inversa. Seguidament és simulen valors uniformes entre 0 i 1 i els hi apliquem la funció inversa trobada obtenint els valors de qualsevol distribució desitjada.

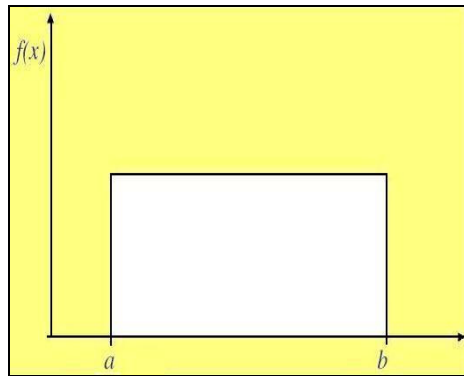


figura 46: exemple de distribució uniforme entra a i b

B.3.4. Distribució Exponencial

Aquesta llei de distribució s'acostuma a utilitzar per representar temps de funcionament o temps en espera. Donat que és una distribució de probabilitat continua, la utilitzarem per definir l'interval de temps entre la creació de dos accidents (i també per a l'actualització). De fet, el que ens interessa és determinar el temps entre l'arribada de dos accidents.

Notació Arena: EXPOnential(Mean[,Stream])

La funció de densitat que depèn d'un paràmetre k és de la forma $f(x) = ke^{-kx}$ i el valor de la mitjana ($1/k$) usat és 25 tenint EXPO(25).

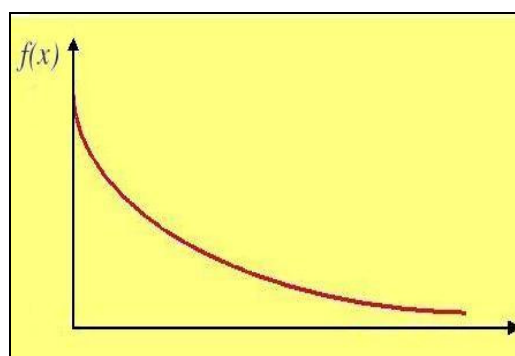


figura 47: exemple de distribució exponencial

ANNEX C. Implementació del model VBA

C.1. Llenguatge SIMAN: blocs VBA

Existeixen diferents paquets de software de simulació per a construir un model d'un sistema particular. Trobem paquets especialitzats en un camp concret i paquets de propòsits generals com és el cas del llenguatge SIMAN.

El llenguatge SIMAN (Simulation and Analysis) es desenvolupa entre el modelatge i l'experimentació. Modela un sistema discret utilitzant la orientació al procés. És a dir, en un model de sistema, s'estudien les entitats que es mouen a través del sistema. Per SIMAN, una entitat és un objecte que es mou en la simulació i que té unes característiques úniques (atributs).

Arena, utilitza aquest llenguatge com a motor per a les simulacions i consta d'infininitat de blocs i opcions.

Els blocs VBA (Visual Basic for Applications) ens permeten escriure codi amb el qual podem extreure tota la informació de les entitats i atributs desitjats ja sigui cap a un fitxer extern o, en el nostre cas, cap a la base de dades.

Tots els mòduls (veure codi implementació), necessiten accedir al model i objecte de dades SIMAN tal i com es mostra a continuació:

```
'Definició de la variable SIMAN global
Dim m As Model
Dim s As SIMAN
'm és el model d'Arena i s és l'objecte de dades SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

C.2. Fitxers kml GoogleEarth

C.2.1. Definició i estructura d'un fitxer KML

KML (Keyhole Markup Language) és un llenguatge de marcat bast en XML per representar dades geogràfiques en tres dimensions en el programa Google Earth.

Els fitxers .kml s'utilitzen per marcar llocs o posicions geogràfiques en forma de coordenades (latitud i longitud).

Per a la representació, s'han usat els següents fitxer kml:

Accidents.kml → conté els accidents que es van produint. Per tant, la informació es va actualitzant.

Flotes.kml (*P, B i A*) → mostra totes les posicions de les flotes dels serveis que trobem en la taula Bases de la BDD. Es genera al principi i no s'actualitza mai.

Moviment.kml (*P,B i A*) → conté una posició del vehicle del servei en qüestió. Aquesta posició es va actualitzant.

Un exemple de l'estructura d'un fitxer kml és la següent:

```
<?xml version='1.0' encoding='UTF-8'?>
<kml xmlns='http://earth.google.com/kml/2.0'>
<Document>
  <Style id='Policia'>
    <IconStyle>
      <scale>1.1</scale>
      <Icon>
        <href>http://maps.google.com/mapfiles/kml/pal5/icon47.png</href>
      </Icon>
    </IconStyle>
  </Style>
  <Placemark><name> VP </name>
    <visibility>1</visibility>
    <styleUrl>#Policia</styleUrl>
    <Point>
      <coordinates>
        2.0957604,41.3803891,0
      </coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

C.2.2. Generació del Moviment al Google Earth

Com s'ha dit abans, els fitxers que corresponen al moviment dels vehicles o a la generació d'accidents són canviants durant la simulació. L'exemple mostrat anteriorment correspon al fitxer *MovimentP.kml* i marca una posició concreta d'un vehicle de Policia. Aquest fitxer, en cada iteració de la línia de moviment, es crea de nou donant lloc a unes noves coordenades (corresponen al node següent).

Per aconseguir aquesta actualització en el Google Earth, necessitem refrescar i sincronitzar el contingut del fitxer. Per fer-ho, utilitzem el fitxer *GeneraMov.kml* el qual s'encarrega de fer aquesta actualització dels fitxers de moviment dels vehicles i el d'accidents. El resultat és un moviment "real" dels vehicles i l'aparició de nous accidents.

Seguidament podem veure la implementació del fitxer (es pot veure el refresc de cada fitxer en la línia RefreshInterval).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">

<Folder>
  <name> Moviment </name>
  <open>1</open>
  <NetworkLink>
    <name>Google Earth - New Network Link</name>
    <Url>
      <href>Accidents.kml</href>
    </Url>
  </NetworkLink>

  <NetworkLink>
    <name>Google Earth - New Network Link</name>
    <Url>
      <href>MovimentP.kml</href>
      <refreshVisibility>1</refreshVisibility>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>0.25</refreshInterval>
    </Url>
  </NetworkLink>

  <NetworkLink>
    <name>Google Earth - New Network Link</name>
    <Url>
      <href>MovimentB.kml</href>
      <refreshVisibility>1</refreshVisibility>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>0.5</refreshInterval>
    </Url>
  </NetworkLink>

  <NetworkLink>
    <name>Google Earth - New Network Link</name>
    <Url>
      <href>MovimentA.kml</href>
      <refreshVisibility>1</refreshVisibility>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>0.25</refreshInterval>
    </Url>
  </NetworkLink>

</Folder>
</kml>
```

C.3. Codi Visual Basic

A continuació es mostra tota la implementació dels blocs VBA que conformen el model de simulació VBA.

```
'Funcions i procediments globals
```

```
Option Explicit
```

```
Private Declare Function comptador Lib "MySQLCorba.dll" Alias
"_comptador@12" (ByVal
host As String, ByVal database As String, ByVal sentencia As String)
As Long
```

```
Private Declare Sub executeUpdate Lib "MySQLCorba.dll" Alias
"_executeUpdate@12" (ByVal host As String, ByVal database As String,
ByVal sentencia As String)
```

```
Private Declare Function executeQueryLong Lib "MySQLCorba.dll" Alias
"_executeQueryLong@12" (ByVal host As String, ByVal database As
String, ByVal sentencia As String) As Long
```

```
Private Declare Sub executeQueryDouble Lib "MySQLCorba.dll" Alias
"_executeQueryDouble@16" (ByVal host As String, ByVal database As
String, ByVal sentencia As String, ByVal res As Double)
```

```
Private Declare Sub executeQueryCampsVert Lib "MySQLCorba.dll" Alias
"_executeQueryCampsVert@20" (ByVal host As String, ByVal database As
String, ByVal sentencia As String, ByVal dimensio As String, ByVal res
As Double)
```

```
Private Declare Sub executeQueryCampsHor Lib "MySQLCorba.dll" Alias
"_executeQueryCampsVert@20" (ByVal host As String, ByVal database As
String, ByVal sentencia As String, ByVal dimensio As String, ByVal res
As Double)
```

C.3.1. Generació d'Accidents

```
Private Sub VBA_Block_1_Fire()
```

```
'Insertem els accidents a la BDD
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "insert into Accidents
values (" & s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) &
" ,1,1,1," & s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("nvi"))
& ", " & s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("nf")) & ", "
```

```

& s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("evi")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("nva")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("eap")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("eab")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("eaa")) & "," &

s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ap")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ab")) & "," &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("aa")) & " "

'Generem dos llocs d'accident per a la representació

Dim idacc As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select idAccident
from Accidents limit 1 ", idacc

If idacc = 1 Then
    executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents
set idNodeAccident=17915 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) "
Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents
set idNodeAccident=45568 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) "
End If

'Generació del fitxer kml per visualitzar la localització dels
Accidents

Dim FSO, f As Object
Dim l As String
Dim i, numAcc As Integer
Dim lat() As Double
Dim lon() As Double
Dim num() As Double

Set FSO = CreateObject("Scripting.FileSystemObject")

Set f = FSO.CreateTextFile("C:\Documents and Settings\guest\
Escritorio\PFCSumalla\GOOGLEEARTH\Accidents.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Accident'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "
<href>http://maps.google.com/mapfiles/kml/pal3/icon41.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)

```

```

l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)

'Busquem la localització exacte (latitud i longitud) de tots els
Accidents insertats a la BDD per a la representació

numAcc = comptador("158.109.69.130", "PFCSumalla", "select count(*)
from Accidents")

ReDim lat(numAcc)
ReDim lon(numAcc)
ReDim num(numAcc)

i = 1

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos Right Join Accidents On Nodos.idNodo =
idNodeAccident order by idAccident", CStr(numAcc), lat(i)
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos Right Join Accidents On Nodos.idNodo =
idNodeAccident order by idAccident", CStr(numAcc), lon(i)
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
idAccident from Accidents order by idAccident", CStr(numAcc), num(i)

For i = 1 To numAcc
l = "<Placemark>"
f.Writeline (l)
l = " <name> A " & i & " </name>"
f.Writeline (l)
l = " <visibility>1</visibility>"
f.Writeline (l)
l = " <styleUrl>#Accident</styleUrl>"
f.Writeline (l)
l = " <Point>"
f.Writeline (l)
l = "     <coordinates>"
f.Writeline (l)
l = "         " & Int(lat(i)) & "." & Right(CStr(Format((lat(i) -
Int(lat(i))), "0.0000000")), Len(CStr(Format((lat(i) - Int(lat(i))),
"0.0000000"))) - 2) & "," & Int(lon(i)) & "." &
Right(CStr(Format((lon(i) - Int(lon(i))), "0.0000000")),
Len(CStr(Format((lon(i) - Int(lon(i))), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "     </coordinates>"
f.Writeline (l)
l = " </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)

Next i

l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

End Sub

```

C.3.2. Actualització dels Accidents

```
Private Sub VBA_Block_17_Fire()
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
'Actualització de les dades de l'Accident (atributs que poden canviar
al rebre nova informació)
```

```
Dim idacc As Double
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select idAccident
from Accidents where (AssigPolicia=0) limit 1 ", idacc
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
numVehicles=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("nvi")) & ", numVictimes=" &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("nf")) &
",hospitalitzacio=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("evi")) & ",numAtrapats=" &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("nva")) &
",idAccident=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("ida")) & " where (AssigPolicia= 0 And
(EstatAccidentPolicia=0 OR EstatAccidentPolicia=1))"
```

```
'necessitem actualitzar ida a la entitat original generada
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
idAccident=" & idacc & " where (idAccident=" &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " )"
```

```
End Sub
```

C.3.3. Generació de les flotes dels serveis

```
Private Sub VBA_Block_9_Fire()
```

```
'Generació del fitxer kml per visualitzar totes les flotes de Policia
que tenim
```

```
Dim FSO, f As Object
Dim l As String
```

```
Set FSO = CreateObject("Scripting.FileSystemObject")
```

```
Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\FlotesP.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
```

```

f.Writeline (l)
l = " <Style id='Policia'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)

l = " <Icon>"
f.Writeline (l)

l = " <href>http://maps.google.com/mapfiles/kml/paddle/blu-
blank.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)

'Número de Flotes de Policia

Dim FlotaPolicia, i As Integer
Dim latp() As Double
Dim lonp() As Double
Dim nump() As Double

FlotaPolicia = comptador("158.109.69.130", "PFCSumalla", "select
count(*) from Bases where idServei=1")

'Busquem la localització exacte (latitud i longitud) de totes les
flotes de Policia per a la representació

ReDim latp(FlotaPolicia + 1)
ReDim lonp(FlotaPolicia + 1)
ReDim nump(FlotaPolicia + 1)

i = 1

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=1 order by idDepot", CStr(FlotaPolicia), latp(i)

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=1 order by idDepot", CStr(FlotaPolicia), lonp(i)

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select idDepot
from Bases where idServei=1 order by idDepot", CStr(FlotaPolicia),
nump(i)

For i = 1 To FlotaPolicia
l = "<Placemark>"
f.Writeline (l)
l = "  <name> </name>"
f.Writeline (l)
l = "  <visibility>1</visibility>"
f.Writeline (l)
l = "  <styleUrl>#Policia</styleUrl>"

```

```

f.Writeline (l)
l = "    <Point>"
f.Writeline (l)
l = "    <coordinates>"
f.Writeline (l)
l = "    " & Int(latp(i)) & "." & Right(CStr(Format((latp(i) -
Int(latp(i))), "0.0000000")), Len(CStr(Format((latp(i) -
Int(latp(i))), "0.0000000"))) - 2) & "," & Int(lonp(i)) & "." &
Right(CStr(Format((lonp(i) - Int(lonp(i))), "0.0000000")),
Len(CStr(Format((lonp(i) - Int(lonp(i))), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "    </coordinates>"
f.Writeline (l)
l = "  </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)

```

```
Next i
```

```

l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

```

'Generació del fitxer kml per visualitzar totes les flotes de Bombers que tenim

```
Set FSO = CreateObject("Scripting.FileSystemObject")
```

```

Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\FlotesB.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = "  <Style id='Bombers'>"
f.Writeline (l)
l = "    <IconStyle>"
f.Writeline (l)
l = "      <scale>1.1</scale>"
f.Writeline (l)
l = "      <Icon>"
f.Writeline (l)
l = "        <href>http://maps.google.com/mapfiles/kml/paddle/red-
blank.png</href>"
f.Writeline (l)
l = "      </Icon>"
f.Writeline (l)
l = "    </IconStyle>"
f.Writeline (l)
l = "  </Style>"
f.Writeline (l)

```

'Número de Flotes de Bombers que tenim a la BDD

```
Dim FlotaBombers, j As Integer
Dim latb() As Double
```

```

Dim lonb() As Double
Dim numb() As Double

FlotaBombers = comptador("158.109.69.130", "PFCSumalla", "select
count(*) from Bases where idServei=2")

'Busquem la localització exacte (latitud i longitud) de totes les
flotes de Bombers per a la representació

ReDim latb(FlotaBombers + 1)
ReDim lonb(FlotaBombers + 1)
ReDim numb(FlotaBombers + 1)

j = 1

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=2 order by idDepot", CStr(FlotaBombers), latb(j)

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=2 order by idDepot", CStr(FlotaBombers), lonb(j)

executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select idDepot
from Bases where idServei=2 order by idDepot", CStr(FlotaBombers),
numb(j)

For j = 1 To FlotaBombers
l = "<Placemark>"
f.Writeline (l)
l = "  <name> </name>"
f.Writeline (l)
l = "  <visibility>1</visibility>"
f.Writeline (l)
l = "  <styleUrl>#Bombers</styleUrl>"
f.Writeline (l)
l = "  <Point>"
f.Writeline (l)
l = "    <coordinates>"
f.Writeline (l)
l = "      " & Int(latb(j)) & "." & Right(CStr(Format((latb(j) -
Int(latb(j))), "0.0000000")), Len(CStr(Format((latb(j) -
Int(latb(j))), "0.0000000"))) - 2) & "," & Int(lonb(j)) & "." &
Right(CStr(Format((lonb(j) - Int(lonb(j))), "0.0000000")),
Len(CStr(Format((lonb(j) - Int(lonb(j))), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "    </coordinates>"
f.Writeline (l)
l = "  </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)

Next j

l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

```



```
'Generació del fitxer kml per visualitzar totes les flotes
d'Ambulàncies que tenim
```

```
Set FSO = CreateObject("Scripting.FileSystemObject")

Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\FlotesA.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Ambulancies'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = " <href>http://maps.google.com/mapfiles/kml/paddle/grn-
blank.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)
```

```
'Número de Flotes d'Ambulàncies que tenim a la BDD
```

```
Dim FlotaAmbulancies
Dim lata() As Double
Dim lona() As Double
Dim numa() As Double
```

```
FlotaAmbulancies = comptador("158.109.69.130", "PFCSumalla", "select
count(*) from Bases where idServei=3")
```

```
'Busquem la localització exacte (latitud i longitud) de totes les
flotes d'Ambulàncies per a la representació
```

```
ReDim lata(FlotaAmbulancies + 1)
ReDim lona(FlotaAmbulancies + 1)
ReDim numa(FlotaAmbulancies + 1)
```

```
j = 1
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=3 order by idDepot", CStr(FlotaBombers), lata(j)
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos Right Join Bases On Nodos.idNodo = idNode
where idServei=3 order by idDepot", CStr(FlotaBombers), lona(j)
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select idDepot
from Bases where idServei=3 order by idDepot", CStr(FlotaAmbulancies),
numa(j)
```

```
For j = 1 To FlotaAmbulancies
l = "<Placemark>"
f.Writeline (l)
l = "  <name> </name>"
f.Writeline (l)
l = "  <visibility>1</visibility>"
f.Writeline (l)
l = "  <styleUrl>#Ambulancies</styleUrl>"
f.Writeline (l)
l = "  <Point>"
f.Writeline (l)
l = "    <coordinates>"
f.Writeline (l)
l = "      " & Int(lata(j)) & "." & Right(CStr(Format((lata(j) -
Int(lata(j))), "0.0000000")), Len(CStr(Format((lata(j) -
Int(lata(j))), "0.0000000"))) - 2) & "," & Int(lona(j)) & "." &
Right(CStr(Format((lona(j) - Int(lona(j))), "0.0000000")),
Len(CStr(Format((lona(j) - Int(lona(j))), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "    </coordinates>"
f.Writeline (l)
l = "  </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)

Next j

l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)
```

```
'si es vol representar hospital(icono en forma de creu roja)
http://maps.google.com/mapfiles/kml/pal3/icon38.png
```

End Sub

C.3.4. Policia

```
Private Sub VBA_Block_2_Fire()
```

```
'Actualitza l'estat de l'accident de Policia a pendent
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentPolicia=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigPolicia= 0 And EstatAccidentPolicia=0)"
```

End Sub

Private Sub VBA_Block_3_Fire()

'Actualitza l'estat de l'accident de Policia a assignat i bloquejem perquè no torni a ser assignat

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentPolicia=2, AssigPolicia=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigPolicia= 0 And EstatAccidentPolicia=1)"
```

End Sub**Private Sub VBA_Block_4_Fire()**

'Actualitza l'estat a ocupat del primer vehicle lliure de Policia que trobem

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
Dim nodeAcc As Double
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) ",
nodeAcc
```

'Agafem la flota que correspon a l'Accident

```
If nodeAcc = 17915 Then
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=0, idAccident=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("ida")) & " where (idDepot=7 And idServei=1 And
Estat=1) limit 1"
```

```
Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set Estat=0, idAccident=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("ida")) & " where (idDepot=3 And idServei=1 And
Estat=1) limit 1"
```

```
End If
```

End Sub**Private Sub VBA_Block_8_Fire()**

'MOVIMENT V.POLICIA

```
Dim va, i, j As Integer
Dim tonode, fromnode, nodeactual As Double
```

```
Dim v As Double
Dim seq As Integer
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
'1.Busquem vehicles de Policia assignats (va) que han de començar el
moviment: aquells vehicles que tenen idaccident assignat &&
idNode=idNodeAccident && idservei=1
```

```
va = comptador("158.109.69.130", "PFCSumalla", "select count(*) from
(Accidents INNER JOIN Vehicles On Accidents.idAccident =
Vehicles.idAccident) LEFT JOIN Accidents as Accidents1 On
Vehicles.idNode = Accidents1.idNodeAccident where
(Vehicles.idServei=1)")
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
idVehicle from (Accidents INNER JOIN Vehicles On Accidents.idAccident
= Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 On
Vehicles.idNode = Accidents1.idNodeAccident where
(((Accidents1.idNodeAccident) is Null) And Vehicles.idServei=1)",
CStr(va), v
```

```
'2.Consultem els valors dels nodes als quals hem d'anar accedint
segons quin sigui l'accident (17915 Ruta 1, 45568 Ruta 7)
```

```
Dim nodeAcc As Double
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & ") ",
nodeAcc
```

```
If nodeAcc = 17915 Then
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.toNode from ArcsGIS Inner Join (Vehicles Inner Join
(Rutes Inner Join Rutes_ArcGIS On Rutes.idRuta =
Rutes_ArcGIS.idRuta) On Vehicles.idVehicle = Rutes.idVehicle) On
ArcsGIS.RID = Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=1 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seq")) & ") ", tonode
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.fromNode from ArcsGIS Inner Join (Vehicles Inner Join
(Rutes Inner Join Rutes_ArcGIS On Rutes.idRuta =
Rutes_ArcGIS.idRuta) On Vehicles.idVehicle = Rutes.idVehicle) On
ArcsGIS.RID = Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=1 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seq")) & ") ", fromnode
```

```
Else: executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.toNode from ArcsGIS Inner Join (Vehicles Inner Join
(Rutes Inner Join Rutes_ArcGIS On Rutes.idRuta =
Rutes_ArcGIS.idRuta) On Vehicles.idVehicle = Rutes.idVehicle) On
ArcsGIS.RID = Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=7 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seq")) & ") ", tonode
```

```

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.fromNode from ArcsGIS Inner Join (Vehicles Inner Join
(Rutes Inner Join Rutes_ArcGIS On Rutes.idRuta =
Rutes_ArcGIS.idRuta) On Vehicles.idVehicle = Rutes.idVehicle) On
ArcsGIS.RID = Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=7 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seq")) & ") ", fromnode

End If

'Generació del fitxer kml per visualitzar el moviment dels vehicles de
Policia

Dim FSO, f As Object
Dim l As String

Set FSO = CreateObject("Scripting.FileSystemObject")

Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentP.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Policia'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "
<href>http://maps.google.com/mapfiles/kml/pal5/icon47.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)

'3.Actualitzem la posició del vehicle en cada moviment (segons si el
node al que volem anar coincideix amb el toNode o el fromNode)

If nodeactual = tonode Then
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
")"
nodeactual = fromnode
Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & tonode & " where (Vehicles.idVehicle =" & v &
")"
nodeactual = tonode
End If

```

```

If (nodeAcc = fromnode Or nodeAcc = tonode) Then

'hem arribat al lloc de l'accident. Actualitzem posició vehicle
Policia al node de l'accident i sortim
    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
    set idNode = " & fromnode & " where (Vehicles.idVehicle = " & v &
    ")"
    s.VariableArrayValue(s.SymbolNumber("llocAcc")) = 1

'sinó continuem
Else: s.VariableArrayValue(s.SymbolNumber("llocAcc")) = 0

End If

'Agafem la latitud i longitud del node actual per a la representació

Dim latmovp As Double
Dim lonmovp As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode where ((Vehicles.idVehicle) = " & v & ")", latmovp
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode where ((Vehicles.idVehicle) = " & v & ")", lonmovp

l = "<Placemark><name> " & v & " </name>"

f.Writeline (l)
l = "    <visibility>1</visibility>"
f.Writeline (l)
l = "    <styleUrl>#Policia</styleUrl>"
f.Writeline (l)
l = "    <Point>"
f.Writeline (l)
l = "        <coordinates>"
f.Writeline (l)
l = "            " & Int(latmovp) & "." & Right(CStr(Format((latmovp -
Int(latmovp)), "0.0000000")), Len(CStr(Format((latmovp -
Int(latmovp)), "0.0000000"))) - 2) & "," & Int(lonmovp) & "." &
Right(CStr(Format((lonmovp - Int(lonmovp)), "0.0000000")),
Len(CStr(Format((lonmovp - Int(lonmovp)), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "        </coordinates>"
f.Writeline (l)
l = "    </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

End Sub

```

Private Sub VBA_Block_5_Fire()

```
'Actualitza l'estat de l'accident de Policia a en servei

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentPolicia=3 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
EstatAccidentPolicia=2)"
```

End Sub**Private Sub VBA_Block_6_Fire()**

```
'Actualitza l'estat de l'accident de Policia a finalitzat

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentPolicia=5 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
EstatAccidentPolicia=3)"

'Com que el vehicle ha acabat la seva intervenció, retorna a la flota
(netegem el fitxer kml pel google earth)

Dim FSO, f As Object
Dim l As String

Set FSO = CreateObject("Scripting.FileSystemObject")
Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentP.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Policia'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "<href>http://maps.google.com/mapfiles/kml/pal5/icon47.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
```

```

f.Writeline (l)
l = " </Style>"
f.Writeline (l)
l = "<Placemark><name> </name>"
f.Writeline (l)
l = "      <visibility>1</visibility>"
f.Writeline (l)
l = "      <styleUrl>#Policia</styleUrl>"
f.Writeline (l)
l = "    <Point>"
f.Writeline (l)
l = "      <coordinates>"
f.Writeline (l)
l = "        0.0,0.0,0"
f.Writeline (l)
l = "      </coordinates>"
f.Writeline (l)

l = "    </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"

f.Writeline (l)

```

End Sub

Private Sub VBA_Block_7_Fire()

'Actualitzacions Vehicle Policia: Retornem l'estat del vehicle al node corresponent a la base segons l'accident

```

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

```

```

Dim nodeAcc As Double
Dim nodeinicial As Double

```

```

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) ",
nodeAcc

```

```

If nodeAcc = 17915 Then
  executeQueryDouble "158.109.69.130", "PFCSumalla", "select
  idNode from Bases where idDepot=7 ", nodeinicial
  executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
  set idNode = " & nodeinicial & " where (idAccident= " &
  s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
  idServei=1 )"

```

```

Else: executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idNode from Bases where idDepot=3 ", nodeinicial

```



```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & nodeinicial & " where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=1 )"
End If

executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=1, idAccident=0 where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=1 )"

End Sub
```

C.3.5. Bombers

Private Sub VBA_Block_10_Fire()

'Actualitza l'estat de l'accident de Bombers a pendent

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentBombers=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigBombers=0 And EstatAccidentBombers=0)"
```

End Sub

Private Sub VBA_Block_11_Fire()

'Actualitza l'estat de l'accident de Bombers a assignat i bloquejem perquè no torni a ser assignat

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set
EstatAccidentBombers=2, AssigBombers=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigBombers=0 And EstatAccidentBombers=1)"
```

End Sub

Private Sub VBA_Block_12_Fire()

'Actualitza l'estat a ocupat del primer vehicle lliure de Bombers que trobem

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

'Agafem la flota que correspon a l'Accident

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=0, idAccident=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("ida")) & " where (idDepot=28 And idServei=2 And
Estat=1) limit 1"
```

End Sub

Private Sub VBA_Block_13_Fire()

'MOVIMENT V.BOMBERS

```
Dim va, i, j As Integer
Dim tonode, fromnode, nodeactual As Double
Dim v As Double
Dim seq As Integer
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

'1.Busquem vehicles de Bombers assignats (va) que han de començar el moviment: aquells vehicles que tenen idaccident assignat && idNode=idNodeAccident && idservei=2

```
va = comptador("158.109.69.130", "PFCSumalla", "select count(*) from
(Accidents INNER JOIN Vehicles On Accidents.idAccident =
Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 On
Vehicles.idNode = Accidents1.idNodeAccident where
(Vehicles.idServei=2)")
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
idVehicle from (Accidents INNER JOIN Vehicles On Accidents.idAccident
= Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 On
Vehicles.idNode = Accidents1.idNodeAccident where
(((Accidents1.idNodeAccident) is Null) And Vehicles.idServei=2)",
CStr(va), v
```

'2.Consultem els valors dels nodes als quals hem d'anar accedint segons quin sigui l'accident (17915 Ruta 7, 45568 Ruta 8)

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.toNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=2 And
```

```

Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqb")) & ") ", tonode

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.fromNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=2 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqb")) & ") ", fromnode

'Generació del fitxer kml per visualitzar el moviment dels vehicles de
bombers

Dim FSO, f As Object
Dim l As String

Set FSO = CreateObject("Scripting.FileSystemObject")
Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentB.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)

l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Bombers'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "<href>http://maps.google.com/mapfiles/kml/pal5/icon57.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)

'3.Actualitzem la posició del vehicle en cada moviment (segons si el
node al que volem anar coincideix amb el toNode o el fromNode)

If nodeactual = tonode Then

    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
")"
    nodeactual = fromnode

Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & tonode & " where (Vehicles.idVehicle =" & v &
")"
    nodeactual = tonode

End If

```

```

Dim nodeAcc As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) ",
nodeAcc

If (nodeAcc = fromnode Or nodeAcc = tonode) Then

'hem arribat al lloc de l'accident. Actualitzem posició vehicle
Bombers al node de l'accident i sortim

    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
")"
    s.VariableArrayValue(s.SymbolNumber("llocAccB")) = 1

'sinó continuem
Else: s.VariableArrayValue(s.SymbolNumber("llocAccB")) = 0

End If

'Agafem la latitud i longitud del node actual per a la representació

Dim latmovb As Double
Dim lonmovb As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode where ((Vehicles.idVehicle) = " & v & ")", latmovp
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode where ((Vehicles.idVehicle) = " & v & ")", lonmovp

l = "<Placemark><name> " & v & " </name>"
f.Writeline (l)
l = "    <visibility>1</visibility>"
f.Writeline (l)
l = "    <styleUrl>#Bombers</styleUrl>"
f.Writeline (l)

l = "    <Point>"
f.Writeline (l)
l = "        <coordinates>"
f.Writeline (l)
l = "            " & Int(latmovb) & "." & Right(CStr(Format((latmovb -
Int(latmovb)), "0.0000000")), Len(CStr(Format((latmovb -
Int(latmovb)), "0.0000000"))) - 2) & "," & Int(lonmovb) & "." &
Right(CStr(Format((lonmovb - Int(lonmovb)), "0.0000000")),
Len(CStr(Format((lonmovb - Int(lonmovb)), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = "        </coordinates>"
f.Writeline (l)
l = "    </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"

```

```
f.Writeline (l)
```

```
End Sub
```

```
Private Sub VBA_Block_14_Fire()
```

```
'Actualitza l'estat de l'accident de Bombers a en servei
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set  
EstatAccidentBombers=3 where(idAccident= " &  
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And  
EstatAccidentBombers=2) "
```

```
End Sub
```

```
Private Sub VBA_Block_15_Fire()
```

```
'Actualitza l'estat de l'accident de Bombers a finalitzat
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents set  
EstatAccidentBombers=5 where(idAccident= " &  
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And  
EstatAccidentBombers=3) "
```

```
'Com que el vehicle ha acabat la seva intervenció, retorna a la flota  
(netegem el fitxer kml pel google earth)
```

```
Dim FSO, f As Object
```

```
Dim l As String
```

```
Set FSO = CreateObject("Scripting.FileSystemObject")
```

```
Set f = FSO.CreateTextFile("C:\Documents and  
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentB.kml", True)
```

```
l = "<?xml version='1.0' encoding='UTF-8'?>"
```

```
f.Writeline (l)
```

```
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
```

```
f.Writeline (l)
```

```
l = "<Document>"
```

```
f.Writeline (l)
```

```
l = " <Style id='Bombers'>"
```

```
f.Writeline (l)
```

```
l = " <IconStyle>"
```

```
f.Writeline (l)
```

```
l = " <scale>1.1</scale>"
```

```
f.Writeline (l)
```

```
l = " <Icon>"
```

```
f.Writeline (l)
```

```
l = "<href>http://maps.google.com/mapfiles/kml/pal5/icon57.png</href>"
```

```

f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)
l = "<Placemark><name> </name>"
f.Writeline (l)
l = "          <visibility>1</visibility>"
f.Writeline (l)
l = "          <styleUrl>#Bombers</styleUrl>"
f.Writeline (l)
l = "    <Point>"
f.Writeline (l)
l = "      <coordinates>"
f.Writeline (l)
l = "        0.0,0.0,0"
f.Writeline (l)
l = "      </coordinates>"
f.Writeline (l)

l = "    </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

```

End Sub

Private Sub VBA_Block_16_Fire()

'Actualitzacions Vehicle Bombers: Retornem l'estat del vehicle al node corresponent a la base segons l'accident

```

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

```

```

Dim nodeinicial As Double

```

```

executeQueryDouble "158.109.69.130", "PFCSumalla", "select idNode from
Bases where idDepot=28 ", nodeinicial
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
idNode = " & nodeinicial & " where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=2 )"

```

```

executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=1, idAccident=0 where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=2 )"

```

End Sub

C.3.6. Ambulàncies

Private Sub VBA_Block_18_Fire()

'Actualitza l'estat de l'accident d'ambulàncies a pendent

Dim m As Model

Dim s As SIMAN

Set m = ThisDocument.Model

Set s = m.SIMAN

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents SET
EstatAccidentAmbulancies=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigAmbulancies=0 And EstatAccidentAmbulancies=0)"
```

End Sub

Private Sub VBA_Block_19_Fire()

'Actualitza l'estat de l'accident d'ambulàncies i bloquejem perquè no torni a ser assignat

Dim m As Model

Dim s As SIMAN

Set m = ThisDocument.Model

Set s = m.SIMAN

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents SET
EstatAccidentAmbulancies=2, AssigAmbulancies=1 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
AssigAmbulancies=0 And (EstatAccidentAmbulancies=1))"
```

End Sub

Private Sub VBA_Block_20_Fire()

'Actualitza l'estat a ocupat del primer vehicle lliure d'ambulàncies que trobem

Dim m As Model

Dim s As SIMAN

Set m = ThisDocument.Model

Set s = m.SIMAN

'Agafem la flota que correspon a l'Accident

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=0, idAccident=" & s.EntityAttribute(s.ActiveEntity,
s.SymbolNumber("ida")) & " where (idDepot=44 And idServei=3 And
Estat=1) limit 1"
```

End Sub

```
Private Sub VBA_Block_21_Fire()
```

```
'MOVIMENT V.AMBULÀNCIES
```

```
Dim va, i, j As Integer
Dim tonode, fromnode, nodeactual As Double
Dim v As Double
Dim seq As Integer
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
'1. Busquem vehicles d'ambulàncies assignats (va) que han de començar
el moviment: aquells vehicles que tenen idaccident assignat &&
idNode=idNodeAccident && idservei=3
```

```
va = comptador("158.109.69.130", "PFCSumalla", "select count(*) from
(Accidents INNER JOIN Vehicles ON Accidents.idAccident =
Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 ON
```

```
Vehicles.idNode = Accidents1.idNodeAccident where
(Vehicles.idServei=3)")
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
idVehicle from (Accidents INNER JOIN Vehicles ON Accidents.idAccident
= Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 ON
Vehicles.idNode = Accidents1.idNodeAccident where
(((Accidents1.idNodeAccident) is Null) And Vehicles.idServei=3)",
CStr(va), v
```

```
'2.Consultem els valors dels nodes als quals hem d'anar accedint
segons quin sigui l'accident (17915 Ruta 7, 45568 Ruta 8)
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.toNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=8 And
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqa")) & ") ", tonode
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.fromNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.idRuta=8 And
```

```
Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqa")) & ") ", fromnode
```

```
'Generació del fitxer kml per visualitzar el moviment de les
ambulàncies
```

```
Dim FSO, f As Object
Dim l As String
```



```

Set FSO = CreateObject("Scripting.FileSystemObject")
Set f = FSO.CreateTextFile("C:\Documents and
Settings\quest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentA.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Ambulancies'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "<href>http://maps.google.com/mapfiles/kml/pal5/icon48.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)

l = " </Style>"
f.Writeline (l)

```

'3.Actualitzem la posició del vehicle en cada moviment (segons si el node al que volem anar coincideix amb el toNode o el fromNode)

```

If nodeactual = tonode Then

    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
")"
    nodeactual = fromnode

Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & tonode & " where (Vehicles.idVehicle =" & v &
")"
    nodeactual = tonode

```

End If

```
Dim nodeAcc As Double
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & ") ",
nodeAcc
```

```
If (nodeAcc = fromnode Or nodeAcc = tonode) Then
```

'hem arribat al lloc de l'accident. Actualitzem posició vehicle
Ambulàncies al node de l'accident i sortim

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
")"
s.VariableArrayValue(s.SymbolNumber("llocAccA")) = 1
```

```

'sinó continuem
Else: s.VariableArrayValue(s.SymbolNumber("llocAccA")) = 0

End If

'Agafem la latitud i longitud del node actual per a la representació

Dim latmova As Double
Dim lonmova As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode WHERE ((Vehicles.idVehicle) = '" & v & "')" , latmova
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode WHERE ((Vehicles.idVehicle) = '" & v & "')" , lonmova

l = "<Placemark><name> " & v & " </name>"
f.Writeline (l)
l = " <visibility>1</visibility>"
f.Writeline (l)
l = " <styleUrl>#Ambulancies</styleUrl>"
f.Writeline (l)
l = " <Point>"
f.Writeline (l)
l = " <coordinates>"
f.Writeline (l)
l = " " & Int(latmova) & "." & Right(CStr(Format((latmova -
Int(latmova)), "0.0000000")), Len(CStr(Format((latmova -
Int(latmova)), "0.0000000"))) - 2) & "," & Int(lonmova) & "." &
Right(CStr(Format((lonmova - Int(lonmova)), "0.0000000")),
Len(CStr(Format((lonmova - Int(lonmova)), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = " </coordinates>"
f.Writeline (l)
l = " </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

End Sub

Private Sub VBA_Block_22_Fire()

'Actualitza l'estat de l'accident d'ambulancies a en servei

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents SET
EstatAccidentAmbulancies=3 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
EstatAccidentAmbulancies=2)"

```

End Sub

Private Sub VBA_Block_23_Fire()

```
'Actualitza l'estat de l'accident d'ambulancies a finalitzat

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

executeUpdate "158.109.69.130", "PFCSumalla", "update Accidents SET
EstatAccidentAmbulancies=5 where(idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
EstatAccidentAmbulancies=3)"

'Com que el vehicle ha acabat la seva intervenció, retorna a la flota
(netegem el fitxer kml pel google earth)

Dim FSO, f As Object
Dim l As String

Set FSO = CreateObject("Scripting.FileSystemObject")
Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentA.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Ambulancies'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = "<href>http://maps.google.com/mapfiles/kml/pal5/icon48.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
l = " </Style>"
f.Writeline (l)
l = "<Placemark><name> </name>"
f.Writeline (l)
l = "      <visibility>1</visibility>"
f.Writeline (l)
l = "      <styleUrl>#Ambulancies</styleUrl>"
f.Writeline (l)
l = "    <Point>"
f.Writeline (l)
l = "      <coordinates>"
f.Writeline (l)
l = "        0.0,0.0,0"
```

```
f.Writeline (l)
l = "          </coordinates>"
f.Writeline (l)
l = "    </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)
```

End Sub

Private Sub VBA_Block_24_Fire()

'Actualitzacions Ambulàncies: Retornem l'estat del vehicle al node corresponent a la base segons l'accident

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

```
Dim nodeinicial As Double
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select idNode from
Bases where idDepot=44 ", nodeinicial
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
idNode = " & nodeinicial & " where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=3 )"
```

```
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=1, idAccident=0 where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=3 )"
```

End Sub

Private Sub VBA_Block_25_Fire()

'MOVIMENT HOSPITAL

```
Dim va, i, j As Integer
Dim tonode, fromnode, nodeactual As Double
Dim v As Double
Dim seq As Integer
```

```
Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN
```

'1.Busquem vehicles d'ambulàncies que han de començar el moviment cap a l'hospital

```
va = comptador("158.109.69.130", "PFCSumalla", "select count(*) from
(Accidents INNER JOIN Vehicles ON Accidents.idAccident =
Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 ON
Vehicles.idNode = Accidents1.idNodeAccident where
(Vehicles.idServei=3)")
```

```
executeQueryCampsVert "158.109.69.130", "PFCSumalla", "select
idVehicle from (Accidents INNER JOIN Vehicles ON Accidents.idAccident
= Vehicles.idAccident) LEFT JOIN Accidents AS Accidents1 ON
Vehicles.idNode = Accidents1.idNodeAccident where
(((Accidents1.idNodeAccident) is Null) And Vehicles.idServei=3)",
CStr(va), v
```

'2.Consultem els valors dels nodes als quals hem d'anar accedint segons l'hospital

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.toNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqh")) & ") ", tonode
```

```
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
ArcsGIS.fromNode from ArcsGIS Inner Join (Vehicles Inner Join (Rutes
Inner Join Rutes_ArcGIS On Rutes.idRuta = Rutes_ArcGIS.idRuta) On
Vehicles.idVehicle = Rutes.idVehicle) On ArcsGIS.RID =
Rutes_ArcGIS.RID where (Rutes_ArcGIS.Sequencia=" &
s.VariableArrayValue(s.SymbolNumber("seqh")) & ") ", fromnode
```

'Generació del fitxer kml per visualitzar el moviment de les ambulàncies

```
Dim FSO, f As Object
```

```
Dim l As String
```

```
Set FSO = CreateObject("Scripting.FileSystemObject")
Set f = FSO.CreateTextFile("C:\Documents and
Settings\guest\Escritorio\PFCSumalla\GOOGLEEARTH\MovimentA.kml", True)
l = "<?xml version='1.0' encoding='UTF-8'?>"
f.Writeline (l)
l = "<kml xmlns='http://earth.google.com/kml/2.0'>"
f.Writeline (l)
l = "<Document>"
f.Writeline (l)
l = " <Style id='Ambulancies'>"
f.Writeline (l)
l = " <IconStyle>"
f.Writeline (l)
l = " <scale>1.1</scale>"
f.Writeline (l)
l = " <Icon>"
f.Writeline (l)
l = " <href>
http://maps.google.com/mapfiles/kml/pal5/icon63.png</href>"
f.Writeline (l)
l = " </Icon>"
f.Writeline (l)
l = " </IconStyle>"
f.Writeline (l)
```

```

l = " </Style>"
f.Writeline (l)

'3.Actualitzem la posició del vehicle en cada moviment (segons si el
node al que volem anar coincideix amb el toNode o el fromNode)

If nodeactual = tonode Then

    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
)" "
    nodeactual = fromnode

Else: executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & tonode & " where (Vehicles.idVehicle =" & v &
)" "
    nodeactual = tonode

End If

Dim nodeAcc As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
idnodeAccident from Accidents where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " ) ",
nodeAcc

If (nodeAcc = fromnode Or nodeAcc = tonode) Then

'hem arribat a l'hospital. Actualitzem posició vehicle Ambulàncies al
node de l'accident i sortim
    executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles
set idNode = " & fromnode & " where (Vehicles.idVehicle =" & v &
)" "
    s.VariableArrayValue(s.SymbolNumber("llocAcH")) = 1

'sinó continuem
Else: s.VariableArrayValue(s.SymbolNumber("llocAcH")) = 0

End If

'Agafem la latitud i longitud del node actual per a la representació

Dim latmova As Double
Dim lonmova As Double

executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.latitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode WHERE (((Vehicles.idVehicle) = '" & v & "'))", latmova
executeQueryDouble "158.109.69.130", "PFCSumalla", "select
Nodos.longitud from Nodos INNER Join Vehicles On Nodos.idNodo =
Vehicles.idNode WHERE (((Vehicles.idVehicle) = '" & v & "'))", lonmova

l = "<Placemark><name> " & v & " </name>"
f.Writeline (l)
l = "    <visibility>1</visibility>"
f.Writeline (l)
l = "    <styleUrl>#Ambulancies</styleUrl>"
f.Writeline (l)

```

```

l = " <Point>"
f.Writeline (l)
l = " <coordinates>"
f.Writeline (l)
l = " " & Int(latmova) & "." & Right(CStr(Format((latmova -
Int(latmova)), "0.0000000")), Len(CStr(Format((latmova -
Int(latmova)), "0.0000000"))) - 2) & "," & Int(lonmova) & "." &
Right(CStr(Format((lonmova - Int(lonmova)), "0.0000000")),
Len(CStr(Format((lonmova - Int(lonmova)), "0.0000000"))) - 2) & ",0"
f.Writeline (l)
l = " </coordinates>"
f.Writeline (l)
l = " </Point>"
f.Writeline (l)
l = "</Placemark>"
f.Writeline (l)
l = "</Document>"
f.Writeline (l)
l = "</kml>"
f.Writeline (l)

```

End Sub

Private Sub VBA_Block_26_Fire()

'Actualitzacions Ambulàncies: Retornem l'estat del vehicle al node corresponent a la base segons l'accident

```

Dim m As Model
Dim s As SIMAN
Set m = ThisDocument.Model
Set s = m.SIMAN

```

Dim nodeinicial As Double

```

executeQueryDouble "158.109.69.130", "PFCSumalla", "select idNode from
Bases where idDepot=44 ", nodeinicial
executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
idNode = " & nodeinicial & " where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=3 )"

```

```

executeUpdate "158.109.69.130", "PFCSumalla", "update Vehicles set
Estat=1, idAccident=0 where (idAccident= " &
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ida")) & " And
idServei=3 )"

```

End Sub

RESUM

El projecte exposat té com a propòsit definir i implementar un model de simulació basat en la coordinació i assignació dels serveis d'emergència en accidents de trànsit.

La definició del model s'ha realitzat amb l'ús de les Xarxes de Petri Acolorides i la implementació amb el software Rockwell Arena 7.0.

El modelatge de la primera simulació ens mostra un model teòric basat en cues mentre que el segon, mostra un model més complet i real gràcies a la connexió mitjançant la plataforma Corba a una base de dades amb informació geogràfica de les flotes i de les rutes.

Com a resultat de l'estudi i amb l'ajuda de GoogleEarth, podem realitzar simulacions gràfiques per veure els accidents generats, les flotes dels serveis i el moviment dels vehicles des de les bases fins als accidents.

RESUMEN

El proyecto expuesto tiene como propósito definir e implementar un modelo de simulación basado en la coordinación y asignación de los servicios de emergencia en accidentes de tránsito.

La definición del modelo se ha realizado usando las Redes de Petri y la implementación mediante el software Rockwell Arena 7.0.

El modelado de la primera simulación nos muestra un modelo teórico basado en colas mientras que el segundo, muestra un modelo más completo y real gracias a la conexión mediante la plataforma Corba a una base de datos con información geográfica de las bases y de las rutas.

Como resultado del estudio y con la ayuda de GoogleEarth podemos realizar simulaciones gráficas para ver los accidentes generados, las flotas de los servicios y el movimiento de los vehículos desde las bases hasta los accidentes.

ABSTRACT

The purpose of the exposed project is to define and implement a model of simulation based on the coordination and allocation of emergency services in traffic accidents.

The definition of the model has been made using Coloured Petri Nets (CPN) and the implementation with the software Rockwell Arena 7.0.

The modeling of the first simulation with the program shows a theoretical model based on queues and the second one, shows a more complete and real model thanks to the connection between the platform Corba and database with a geographic information of the positions of depots and routes.

As a result of the study and with the aid of GoogleEarth we can make graphical simulations to see the accidents generated, the depots of the services and the movement of the vehicles from depots to the accidents.