

Anàlisi de Requeriments per un Model de Xarxa de Telecomunicacions amb aplicació des d'un Sistema d'Informació Geogràfica

Santiago Payà i Miralta

Enginyeria Tècnica Informàtica de Gestió
Universitat Oberta de Catalunya

Tutor del Treball
José Antonio Raya Martos

18 de Juny de 2004

Copyright i Llicència

Copyright (c) 2004 Santiago Payà i Miralta.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índex

I	Memòria	1
1	Introducció	2
1.1	Justificació i Context	2
1.2	Objectius	2
1.3	Enfocament i Mètode	3
1.4	Planificació	3
1.5	Productes	3
1.6	Descripció dels capítols	3
2	Recollida de requeriments	5
2.1	Informació inicial	5
2.2	Context del programari	6
2.2.1	Model de domini	6
2.2.2	Model de negoci	7
2.3	Guions	8
2.4	Actors	8
2.5	Casos d'ús	8
2.6	Interfície d'usuari	12
2.6.1	Perfils d'usuari	12
2.6.2	Descripció de les tasques	12
2.6.3	Requisits d'usabilitat	13
3	Anàlisi de requeriments	14
3.1	Paquets	14
3.2	Revisió dels casos d'ús	14
3.3	Classes d'anàlisi	15
3.3.1	Classes d'entitat	16
3.3.2	Relacions	18
3.3.3	Diagrames de col·laboració	19
3.3.4	Diagrama estàtic d'anàlisi	23
3.4	Especificació formal dels casos d'ús	23
3.4.1	Diagrames de seqüències	27
3.4.2	Diagrames d'estats	27
3.5	Interfície d'usuari	27
3.6	Revisió dels paquets	27

4	Disseny	36
4.1	Arquitectura	36
4.1.1	Configuració de la xarxa	37
4.1.2	Punt de partida dels subsistemes	37
4.2	Casos d'Ús	38
4.3	Diagrama Estàtic de Disseny	42
4.3.1	Disseny de la persistència	42
4.3.2	Subsistemes	47
4.4	Interfície	50
5	Valoració econòmica	55
6	Conclusions	56
7	Glossari	57
7.1	Introducció	57
7.2	Elements lineals	57
7.2.1	Canalització	57
7.2.2	Conductes	58
7.2.3	Cables	58
7.2.4	Fibres	58
7.3	Elements puntuals	59
7.3.1	Registre	59
7.3.2	Caixa d'empulaments	59
7.3.3	Empulaments	60
II	Annexes	63
8	Pla de treball	64
8.1	Descripció del Treball Fi de Carrera	64
8.1.1	Objectius Generals i Específics	64
8.1.2	Planificació	65
8.1.3	Model i Mètode	65
8.1.4	Recursos	66
9	Recursos de programari	67
9.1	PostgreSQL	67
9.2	JUMP	69
10	GNU Free Documentation License	71
1.	APPLICABILITY AND DEFINITIONS	71
2.	VERBATIM COPYING	73
3.	COPYING IN QUANTITY	73
4.	MODIFICATIONS	73
5.	COMBINING DOCUMENTS	75
6.	COLLECTIONS OF DOCUMENTS	75
7.	AGGREGATION WITH INDEPENDENT WORKS	76

8. TRANSLATION	76
9. TERMINATION	76
10. FUTURE REVISIONS OF THIS LICENSE	76
ADDENDUM: How to use this License for your documents	77
11 Codi implementat	78
11.1 Model de Dades	78
11.2 Client GIS	93

Índex de figures

2.1	Exemple de graf	6
2.2	Classes de domini	6
2.3	Casos d'ús de domini	7
2.4	Col·laboració de domini	7
2.5	Casos d'ús	9
3.1	Paquets	14
3.2	Xarxa de canalitzacions i registres	15
3.3	Xarxa de cables i caixes	16
3.4	Primera revisió de les Classes	17
3.5	Herències	18
3.6	Segona revisió de les Classes	20
3.7	Diagrama de col·laboració d'Editar canalització	21
3.8	Diagrama de col·laboració d'Editar conducte	21
3.9	Diagrama de col·laboració d'Editar registre	22
3.10	Diagrama de col·laboració d'Editar cable	22
3.11	Diagrama de col·laboració d'Editar caixa	24
3.12	Diagrama de col·laboració d'Editar empiulament	24
3.13	Diagrama de col·laboració de Plotejar ruta	28
3.14	Diagrama estàtic d'anàlisi	29
3.15	Diagrama de seqüències d'Editar canalització	30
3.16	Diagrama de seqüències d'Editar conducte	30
3.17	Diagrama de seqüències d'Editar registre	31
3.18	Diagrama de seqüències d'Editar cable	31
3.19	Diagrama de seqüències d'Editar caixa	32
3.20	Diagrama de seqüències d'Editar empiulament	32
3.21	Diagrama de seqüències de Plotejar ruta	33
3.22	Diagrama d'Estats	33
3.23	Interfície de la classe EditarCanalitzacio	33
3.24	Interfície de la classe EditarConducte	33
3.25	Interfície de la classe EditarRegistre	34
3.26	Interfície de la classe EditarCable	34
3.27	Interfície de la classe EditarCaixa	34
3.28	Interfície de la classe EditarEmpiulament	35
3.29	Interfície de la classe PlotejarRuta	35
3.30	Revisió dels paquets	35

4.1	Diagrama dels subsistemes de l'arquitectura	37
4.2	Nodes de comunicació de l'arquitectura	37
4.3	Diagrama estàtic de disseny en primera revisió	43
4.4	Model d'Entitat — Relació	44
4.5	Diagrama estàtic de disseny final	49
4.6	Interfície de la classe DialegEditarCanalitzacio	51
4.7	Interfície de la classe DialegEditarConducte	51
4.8	Interfície de la classe DialegEditarRegistre	51
4.9	Interfície de la classe DialegEditarCable	52
4.10	Interfície de la classe DialegEditarCaixa	52
4.11	Interfície de la classe DialegEditarEmpiulament	52
4.12	Interfície de la classe DialegPlotejarRuta	53
4.13	Interfície de dibuix de JUMP	53
4.14	Interfície de PostGISDriver	54
4.15	Interfície de comandes ClientGIS	54
4.16	Interfície de PostgreSQL	54
7.1	Esquema transversal de canalització	57
7.2	Cable de fibra òptica	58
7.3	Fibra òptica	59
7.4	Registre	59
7.5	Caixa d'empulaments	60
7.6	Empiulament	60
8.1	Calendari	66
9.1	Captura de la pàgina web de PostgreSQL	68
9.2	Captura de la pàgina web de JUMP	69

Resum

Aquest projecte és el Treball de Fi de Carrera d'Enginyeria Tècnica en Informàtica de Gestió de l'autor a la Universitat Oberta de Catalunya. La idea principal és la creació d'un model d'una xarxa de telecomunicacions per la seva gestió des d'un programari d'informació geogràfica (GIS). Inicialment, i en el que és l'abast d'aquest treball, es tracta de modelitzar una xarxa simple de fibra òptica similar a les que es realitzen actualment per la connexió directa de clients amb aquesta tecnologia. Les tasques que s'han desenvolupat han sigut les següents:

- Recollida dels requeriments més importants a tenir en compte en la gestió d'una xarxa de telecomunicacions.
- Anàlisi dels requeriments recollits.
- Disseny d'una solució integrada que permeti la funcionalitat desitjada.

En tot aquest procés s'ha seguit un model orientat a objectes amb Unified Modeling Language (UML) i el mètode Rational Unified Process que el desenvolupa. Se seguirà un cicle de vida iteratiu i incremental, de manera que el projecte pugui anar creixent i millorant.

Part I

Memòria

Capítol 1

Introducció

1.1 Justificació i Context

Aquest treball de fi de carrera neix de la inquietud per proporcionar una eina de treball eficient que doni suport a determinades tasques que es desenvolupen o amb les que l'autor té relació en la seva feina dins d'una operadora de telecomunicacions. Igualment es tracta d'aconseguir integrar el que actualment és suportat per diferents eines informàtiques o fins i tot manualment, en una única interfície que aporti els beneficis inherents a la utilització d'una única estructura de dades i d'una interfície comuna.

El primer pas per aconseguir-ho és evidentment posar sobre paper quin és l'abast que es vol arribar a donar a aquest aplicatiu, detectar els requeriments i dissenyar una solució. Com que les possibilitats són pràcticament il·limitades el que es fa és limitar el domini a les funcionalitats més bàsiques i crear una primera versió del disseny a partir d'aquestes. L'esperança és que sobre aquesta primera versió es pugui seguir afegint noves funcionalitats més avançades, per tant és necessari tenir en compte que en el futur determinades parts del treball seran tornades a analitzar i possiblement refetes.

El plantejament general del treball és doncs analitzar una petita part dels requisits, els més importants de cara a posar en marxa el sistema dins de l'organització, i fer el disseny corresponent. Es tracta d'aplicar un desenvolupament iteratiu i incremental.

Paral·lelament al treball s'ha desenvolupat un petit prototipus[9] que permet únicament demostrar que és possible la connexió dels diferents aplicatius que farà servir el programari fruit d'aquest treball i la seva interrelació funcional; en concret s'ha connectat un sistema gestor de bases de dades (DBMS) amb un sistema d'informació geogràfica (GIS) i s'han implementat algunes rutines al sistema de bases de dades. Així doncs, més que un prototipus es tracta d'un experiment que tregui incertesa a la possibilitat d'implementar el disseny resultant amb l'acoblament de les eines que està previst utilitzar.

1.2 Objectius

Tal com es comenta a l'apartat anterior l'objectiu bàsic del treball és arribar a tenir el disseny d'una aplicació que permeti gestionar una xarxa de telecomunicacions de fibra òptica en les seves funcionalitats més bàsiques, és a dir que reculli aquelles característiques que han de fer el model de la xarxa coherent i que permeten la seva creació i edició a través d'un sistema GIS, i també permeten mantenir

una mínima gestió dels serveis de telecomunicacions.

Com objectiu secundari hi ha el desenvolupament del prototipus presentat, però sempre com una experimentació de les funcionalitats dissenyades, i en últim terme com un exemple aplicat.

1.3 Enfocament i Mètode

El treball té doncs dues branques principals, d'una banda es dissenya un model de base de dades que doni suport a la inventari de la xarxa i que permeti extreure'n la funcionalitat desitjada fàcilment des d'un client GIS. D'altra banda el disseny de les rutines i interfície del client GIS que doni resposta a les funcionalitats requerides.

Per aconseguir-ho se segueix el model orientat a objectes, de manera que els diferents elements físics existents en una xarxa real tenen la seva representació dins del programari. L'aplicació del mètode estàndard de l'Unified Software Development Process (UP) desenvolupat per Rational Software Corporation permet plasmar els requeriments de forma coherent amb el desenvolupament del model de dades. Tot el procés fa servir la notació de l'Unified Modeling Language (UML).

No es fa servir eines CASE (Computer Aided Software Engineering) directament, tot i que les aplicacions utilitzades permeten la traducció cap a codi dels desenvolupaments fets. En concret de l'aplicació que es fa servir per traçar els diagrames [18] pot extreure's codi SQL (Standard Query Language) per crear l'esquema d'una base de dades o també codi font en diferents llenguatges a partir d'un diagrama de classes, igualment l'aplicatiu que s'utilitza per l'esboç de les interfícies d'usuari [19]. Tot el document del treball s'ha realitzat amb \LaTeX [10].

1.4 Planificació

La planificació prevista segueix els requeriment d'entregues de l'assignatura del treball. Pot consultar-se la planificació al document "Pla de Treball" en el Capítol 8 als Annexes.

1.5 Productes

El producte final del treball és l'especificació del disseny de la base de dades i del client del programari GIS gestor d'aquestes dades.

En el cas favorable també està previst com entregable el prototipus que s'ha d'entendre com un exemple aplicat, però cal tenir present que no és l'objectiu principal del treball.

1.6 Descripció dels capítols

Els capítols segueixen un ordre cronològic en el desenvolupament de l'objectiu. Així primer hi ha la recollida de requeriments, l'anàlisi d'aquests requeriments i el disseny de la solució. Cal esmentar que hi ha altres apartats que tenen relació amb tots els capítols com per exemple el Glossari o els Annexes.

La descripció dels diferents capítols és la següent:

- Recollida de requeriments: En aquest capítol es fa la recopilació dels requeriments que es fan a l'aplicatiu a desenvolupar. Es defineix l'abast i es justifica el domini que s'escull. Hi ha diagrames de classes i de casos d'ús.
- Anàlisi de requeriments: Anàlisi dels requeriments detectats en el capítol anterior. S'aconsegueixen els diagrames de classes i de col·laboració dels casos d'ús revisats. També el diagrama estàtic de l'anàlisi i les interfícies d'usuari.
- Disseny: Disseny de la solució tecnològica escollida, de manera que permeti assolir els requeriments presentats pels usuaris de la manera més eficient. Es presenta el programari que està previst utilitzar de manera que el disseny s'hi adapti.
- Valoració econòmica: Impacte econòmic de la implementació i implantació del model en una companyia.
- Conclusions: Breu recull de les impressions i valoració dels objectius aconseguits amb el present treball.

Capítol 2

Recollida de requeriments

2.1 Informació inicial

Una operadora de telecomunicacions es dedica a connectar els seus clients directament amb fibra òptica a través de la seva xarxa. Per això construeix una xarxa de canalitzacions per la que estén els cables de fibra òptica. El procés es pot resumir com:

1. El departament comercial contacta el futur client i demana una viabilitat de connexió als departaments tècnic i financer. Si la viabilitat és positiva es llença un projecte de connexió del client.
2. El departament tècnic pren la responsabilitat de connectar el client, de manera que demana les preceptives llicències de construcció de la xarxa als organismes competents, normalment ajuntaments, i porta a terme la construcció de la xarxa.
3. La xarxa estàndard està formada per una canalització d'obra civil amb conductes i registres d'accés. Aquesta és fixa. Posteriorment es fa l'estesa de cable de fibra òptica per l'interior del conductes de manera que es pugui connectar el client des d'una de les centrals de comunicacions de l'operadora. Dins del client es col·loca un armari repartidor, de funcionalitat equivalent als típics repartidors de parells de coure que es fan servir per la xarxa de telefonia bàsica.
4. Com últim pas es fan les connexions de les fibres òptiques creant una ruta, generalment de dues fibres una per transmissió i l'altra per recepció, entre el node i el client. A través d'aquesta ruta única i directa es passen els serveis contractats pel client (veu, dades, internet, ...). És interessant obtenir una ruta sobre el plànol, de manera que sigui relativament fàcil i visualment ràpid el poder esbrinar per quines canalitzacions passa un determinat servei.

L'àmbit que es desitja gestionar pel programari d'aquest treball és la construcció i inventariat de la xarxa, de manera que es pugui generar la xarxa des de l'execució de la canalització fins la creació de les rutes d'emplaments de fibres òptiques, i editar aquestes dades. També es desitja una mínima funcionalitat de control sobre la xarxa.

Cal tenir present que les funcionalitats de gestió econòmica, de contractació, de permisos ... ja estan suportades per altres aplicacions dins de l'operadora.

2.2 Context del programari

2.2.1 Model de domini

L'entorn en el que es fa servir el programari és un departament de construcció i un departament de gestió de fibres òptiques d'una empresa de telecomunicacions que desenvolupa una xarxa de fibra òptica. En aquest domini els usuaris tracten directament amb la xarxa, construint-la i modificant-la.

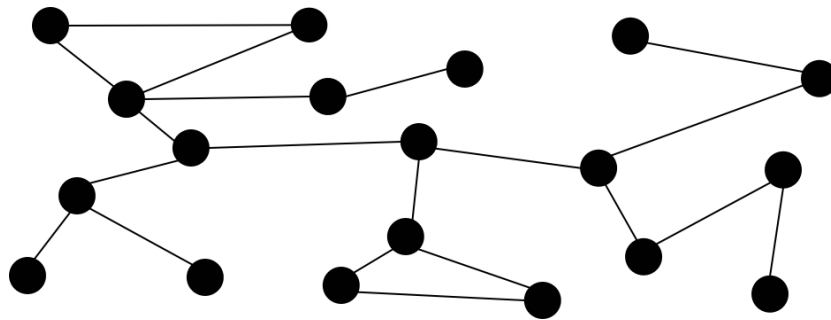


Figura 2.1: Exemple de graf

Com primera aproximació pot considerar-se la xarxa esquemàticament com un graf, figura descriptiva de les relacions entre diferents elements i que consta de parts lineals i punts d'unió que són els vèrtexs. Els trams lineals poden dividir-se als vèrtexs en més trams o no, configurant una xarxa.

Els elements amb els que treballen habitualment els esmentats departaments són els representats pel següent diagrama de classes de domini, elements que es descriuen en el Glossari:

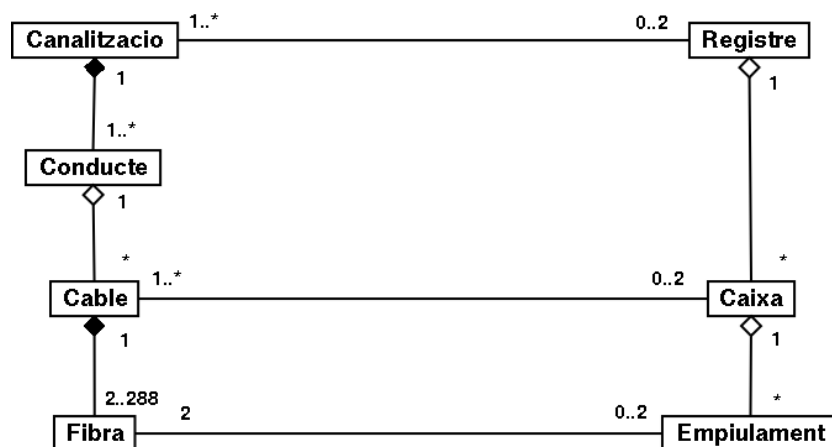


Figura 2.2: Classes de domini

En aquest model de domini es considera una xarxa mínimament operativa. Així doncs no es consideren registres aïllats (sense cap canalització que el connecti) i així es declara amb les cardinalitats de la relació entre **Registre** i **Canalització**. En el cas de connector de fibres **Empiulament** ha de connectar dues fibres exactament, doncs tal com es descriu en el Glossari les fibres es connecten encarant-les una a una. Igualment el nombre de fibres per cable varia de 2 a 288, que són les mides habitualment mínima i màxima de fabricació.

2.2.2 Model de negoci

Per definir el model de negoci cal tenir present els diferents actors que participen en l'àmbit que es vol donar al programari, sense oblidar que algunes de les funcionalitat esmentades ja són suportades per aplicatius ja existents.

D'aquesta manera es detecten els següents actors i casos d'ús:

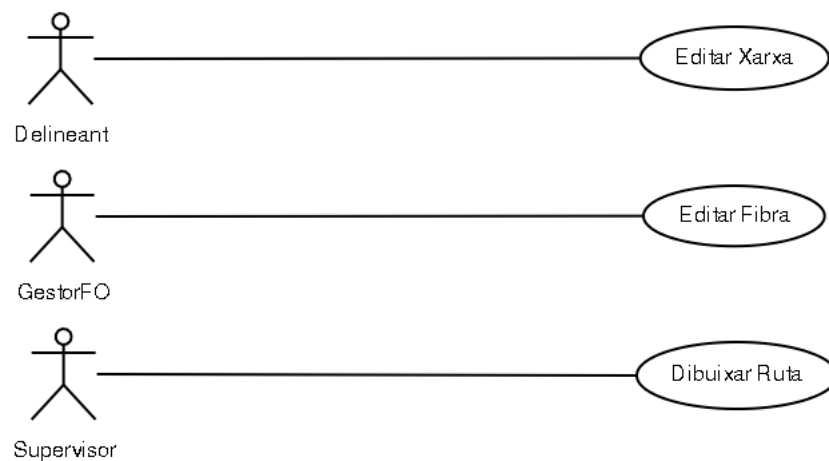


Figura 2.3: Casos d'ús de domini

La relació entre els casos d'ús detectats i els objectes definits es fa a través del diagrama de col·laboració simplificat.

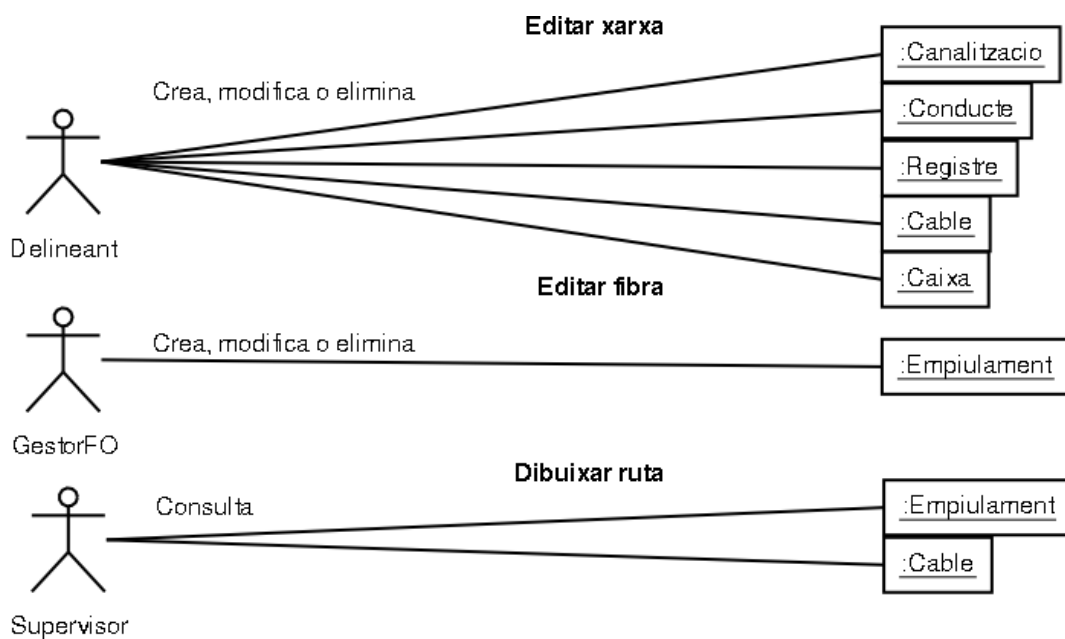


Figura 2.4: Col·laboració de domini

2.3 Guions

Es detecten tres guions:

Guió del delineant: Introdueix les dades de la xarxa que recull de diferents fonts i les plasma a l'aplicació. Dels tècnics que gestionen l'obra (caps d'obra) introdueix la major part dels traçats, característiques i detalls tant de l'obra civil com de l'estesa dels cables i ubicació de les caixes. També recull la informació dels 'as-builts' que donen fe de les tasques realitzades per part de les contractes. Igualment fa modificacions a les dades de la xarxa quan així s'escau, com per exemple en les afectacions de serveis que modifiquen la traçada o en els canvis de cables.

Guió del gestor de fibres òptiques (FO): Introdueix les rutes de fibra per connectar els diferents clients, de manera que les fibres arribin directament del node o central de l'operadora fins el client. Per aconseguir-ho indica els empiulaments a realitzar en els diferents punts de la xarxa (caixes d'empiulaments).

Guió del supervisor: Traça sobre el plànol una ruta de fibres d'un servei o client determinat. D'aquesta manera pot analitzar l'impacte d'algun fet que hagi succeït sobre el servei proporcionat o, en l'hipotètic cas d'un tall de servei, pot calcular quin és el punt més probable de tall de les fibres gràcies a la informació de distància fins el tall que esbrina des del node amb un aparell especial.

2.4 Actors

Els actors per tant són tres: *delineant*, *gestorFO* (gestor de fibres òptiques) i *supervisor*. Els tres són usuaris finals i no existeix cap relació d'especialització, doncs les tasques que realitza cada un no són compartides, no hi ha cap cas d'ús que puguin fer més de dos actors.

El *delineant* té cinc papers doncs introdueix les dades de la xarxa i això inclou traçar la canalització, els conductes, els cables, col·locar els registres i les caixes d'empiulaments. El *gestorFO* i el *supervisor* només realitzen un paper.

2.5 Casos d'ús

El diagrama de casos d'ús (Figura 2.5) mostra aquests papers dels diferents actors. La descripció textual dels diferents casos d'ús és:

1. Cas d'ús **Editar canalització**

Resum de funcionalitat: Edita la canalització, afegint, modificant o eliminant la traçada o les dades relacionades.

Paper dins del treball de l'usuari: És un cas d'ús principal pel *delineant*.

Actors: *delineant*

Casos d'ús relacionats: **Editar registre, Editar conducte, Dades model**

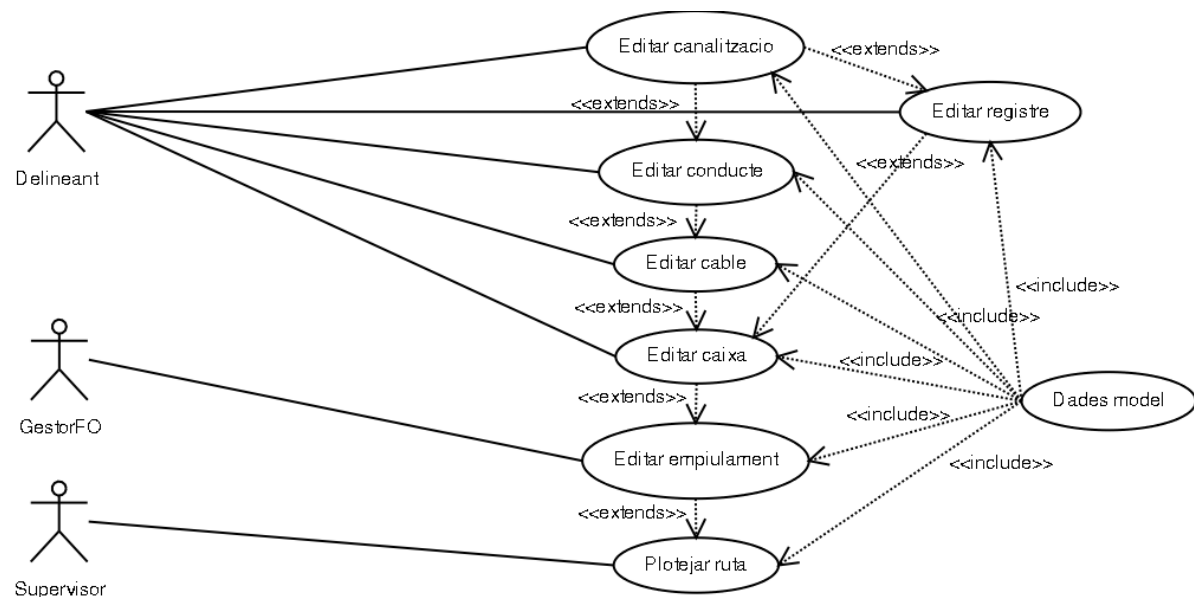


Figura 2.5: Casos d'ús

Precondició: La canalització no existeix o les dades associades són incorrectes o la seva traçada és incorrecte o ja no és actual.

Postcondició: La canalització existeix i discorre per la traçada correcta i les seves dades associades són correctes o ja no existeix.

Descripció: El *delineant* traça sobre el plànol una nova canalització i hi afegeix les dades relacionades (atributs) o modifica la traçada de la canalització sobre el plànol o modifica els atributs o elimina la canalització directament. Els atributs són un codi de projecte, el tipus de canalització (en rasa, en galeria, ...) i el contractista que l'ha executada.

Alternatives i excepcions: En el cas d'eliminar una canalització totes les dades i els elements del model que en depenen també són eliminats.

2. Cas d'ús **Editar conducte**

Resum de funcionalitat: Edita els conductes que formen part de la canalització, afegint, modificant o eliminant el nombre de conductes i les seves característiques.

Paper dins del treball de l'usuari: És un cas d'ús principal pel *delineant*.

Actors: *delineant*

Casos d'ús relacionats: **Editar canalització, Editar cable**

Precondició: El tram de canalització no conté cap conducte o els conductes que conté no són correctes en nombre o en atributs.

Postcondició: Els conductes del tram de canalització són correctes en nombre i en atributs.

Descripció: El *delineant* selecciona un o més trams de canalització i pot editar el nombre i els atributs dels conductes que hi ha a cada un dels trams. Els atributs dels conductes són el tipus de conducte (40mm, 125mm, ...) i el propietari o titular del conducte (propri de l'empresa operadora, de propietat municipal, de reserva, ...).

Alternatives i excepcions: En el cas d'eliminar un conducte totes les dades i els elements del model que en depenen també són eliminats. Un conducte només pot ser editat a partir de la canalització a la que pertany, és a dir que no es poden crear conductes sense una canalització associada.

3. Cas d'ús **Editar registre**

Resum de funcionalitat: Edita els registres que formen part de la canalització, afegint, modificant la seva ubicació o eliminant-lo o modificant els seus atributs.

Paper dins del treball de l'usuari: És un cas d'ús principal pel *delineant*.

Actors: *delineant*

Casos d'ús relacionats: **Editar canalització, Editar caixa**

Precondició: Un punt entre dos trams de canalització o un final de canalització on ha d'anar ubicat un registre no presenta un registre o la ubicació o els atributs del registre són incorrectes.

Postcondició: Al punt se li ha afegit el registre o el registre és correcte en posició i atributs.

Descripció: El *delineant* ha de col·locar els registres en els punts de la xarxa on existeixen. També pot ocórrer que calgui eliminar un registre o desplaçar-lo. Igualment les característiques del registre poden variar. Els atributs del registre són un codi de projecte, el tipus de registre (A, B, C, ...) i el contractista que l'ha executat.

Alternatives i excepcions: En el cas d'eliminar un registre totes les dades i els elements del model que en depenen també són eliminats. Cal tenir present que els registres són elements d'unió dels trams de canalització, per tant han de situar-se en els extrems d'aquests trams. Això generalment s'aconsegueix amb les facilitats topològiques que donen els programaris GIS.

4. Cas d'ús **Editar cable**

Resum de funcionalitat: Edita els cables que es col·loquen per dins dels conductes, afegint, modificant la seva ubicació o conducte en el que estan col·locats o eliminant-lo o modificant els seus atributs.

Paper dins del treball de l'usuari: És un cas d'ús principal pel *delineant*.

Actors: *delineant*

Casos d'ús relacionats: **Editar conducte, Editar caixa**

Precondició: Un o més trams de canalització no contenen en els seus tubs un cable, o les característiques d'un cable no són correctes o passa per conductes erronis.

Postcondició: L'estesa del cable passa pels conductes correctes i els seus atributs són correctes.

Descripció: El *delineant* selecciona els trams de canalització per on es realitza una estesa de cable. Per cada tram selecciona el tub per on es passa el cable. Els atributs del cable no varien en una estesa i són un codi de projecte, el nom del cable, el nombre de fibres òptiques que té el cable (16, 64, 96, 144, 288) i el contractista instal·lador que ha fet l'estesa. Igualment el *delineant* pot seleccionar un o més trams de canalització i descobrir els cables que passen per cada tram per modificar les seves dades o eliminar-lo.

Alternatives i excepcions: En el cas d'eliminar un cable totes les dades i els elements del model que en depenen també són eliminats.

5. Cas d'ús **Editar caixa**

Resum de funcionalitat: Edita les caixes que es col·loquen dins del registre per connectar els cables que hi passen. Es crea, modifica les dades o elimina una caixa.

Paper dins del treball de l'usuari: És un cas d'ús principal del *delineant*.

Actors: *delineant*

Casos d'ús relacionats: **Editar cable, Editar registre**

Precondició: En un registre on hi arriba com a mínim un cable no existeix una caixa d'empulaments o hi ha un cable al registre que es connecta a una caixa existent.

Postcondició: Els cables que es connecten a través de la caixa són els correctes i els atributs de la caixa són els correctes.

Descripció: El *delineant* sel·lecciona el registre on ha de col·locar o editar la caixa d'empulaments. Dins del registre hi pot haver diferents cables, per tant sel·lecciona quins són els cables que entren a la caixa. Els atributs de la caixa són un codi de projecte, el nom de la caixa, el tipus de caixa (32 empulaments, 128 empulaments, ...) i el contractista instal·lador que l'ha col·locat.

Alternatives i excepcions: En el cas d'eliminar una caixa totes les dades i els elements del model que en depenen també són eliminats. Els cables que hi passaven quedaran desconectats. Les caixes es col·loquen dins de registres, no es poden crear caixes sense un registre associat.

6. Cas d'ús **Editar empulament**

Resum de funcionalitat: Edita l'empulament que connecta dues fibres, creant-lo o eliminant-lo.

Paper dins del treball de l'usuari: És un cas d'ús principal del *gestorFO*

Actors: *gestorFO*

Casos d'ús relacionats: **Editar caixa, Plotejar ruta**

Precondició: Dins d'una caixa d'empulaments no existeix la connexió entre dues fibres o la connexió existent no és correcta.

Postcondició: L'empulament de les dues fibres és correcte.

Descripció: El *gestorFO* sel·lecciona el registre on hi ha la caixa on ha de fer els empulaments. Es presenten les fibres que arriben a la caixa de cada un dels cables que hi entren i els empulaments ja existents. El *gestorFO* edita els empulaments creant-ne o eliminant-ne. Cada empulament té com atribut el nom de la ruta.

Alternatives i excepcions: Segons el tipus de caixa i la disposició de les safates existeix una limitació al nombre de tubs que pot haver per safata. Això provoca que pugui ser físicament impossible connectar dues fibres que siguin de tubs col·locats en diferents safates per empulaments anteriors.

7. Cas d'ús **Plotejar ruta**

Resum de funcionalitat: Traça sobre el plànol la ruta que connecta la central de l'operadora amb el client.

Paper dins del treball de l'usuari: És un cas d'ús principal del *supervisor*.

Actors: *supervisor*

Casos d'ús relacionats: **Editar empiulament**

Precondició: Existeix la necessitat de conèixer sobre el plànol la traçada completa d'una ruta que connecta un client des d'una central.

Postcondició: La ruta és dibuixada sobre el plànol.

Descripció: El *supervisor* sel·lecciona una fibra determinada de la que coneix el client a qui es dóna servei en un dels cables que arriba a la central. La ruta completa de les canalitzacions per on passa la fibra connectada des de la central fins el client és plotjada sobre el plànol.

Alternatives i excepcions: no detectades.

2.6 Interfície d'usuari

Es descriu a continuació la funcionalitat que ha de tenir la interfície d'usuari.

2.6.1 Perfils d'usuari

Tots els usuaris són avançats en la utilització de programari i coneixen perfectament la seva feina. La utilització de l'aplicatiu serà constant en els casos del *delineant* i del *gestorFO*, i potser menor en el cas del *supervisor*.

2.6.2 Descripció de les tasques

Edició de la xarxa d'obra civil i de cables

Aquesta tasca correspon a les edicions de canalització, conducte, cable, registre i caixa que porta a terme el *delineant*, es tracta en definitiva de crear i mantenir la xarxa tal qual existeix en el món real, de manera que per sota s'hi puguin afegir les rutes de fibres i els serveis que hi passen.

El *delineant* treballa directament només contra canalitzacions i contra registres, la resta d'elements del model com són els conductes, els cables, les caixes, ... no apareixen com a tals en el plànol sinó que se suposen interns a cada tram de canalització o a cada registre. Per tant la manera de treballar del *delineant* és sempre contra una sel·lecció de canalitzacions i/o registres als que podrà anar afegint, modificant o eliminant els elements del model que en depenguin.

La interfície per tant és, un cop sel·leccionat un o més registres i/o canalitzacions, un diàleg en el que pot modificar-se els atributs dels registres i canalitzacions i també afegir-se, modificar-se o eliminar-se conductes a les canalitzacions i caixes als registres. També és possible afegir, modificar o eliminar els cables que passin per un determinat conducte i les caixes que hi hagi en un determinat registre. Per fi també es pot afegir o eliminar els cables que entren en una determinada caixa.

D'aquesta manera el *delineant* crea línies i punts directament amb l'eina GIS i en sel·leccionar-los i obrir el diàleg corresponent els pot afegir els atributs i els elements del model relacionats, construint i mantenint una xarxa completa d'obra civil i de cables.

La freqüència és de 100 elements lineal i 100 elements puntuals al dia. Totes les entrades i sortides són a través de la pantalla i aplicatiu client GIS. El *delineant* necessita per la seva feina la cartografia de la població. Els problemes i errors més freqüents que es troben són la falta d'informació detallada en el moment de traspassar-la a l'aplicatiu, el que obliga a revisions periòdiques.

Edició d'empulaments

Aquesta tasca correspon al manteniment de la xarxa de fibra òptica que és interna a la xarxa de cables que ha realitzat el *delineant*. El *gestorFO* és l'encarregat d'aquesta tasca i la realitza al sel·leccionar cada registre i modificar els empulaments que hi ha en la caixa o caixes que hi hagi dins del registre.

La interfície per tant és, un cop sel·leccionat el registre, un diàleg en el que es mostren les caixes presents amb els cables que hi arriben a cada caixa. En sel·leccionar una caixa es poden eliminar o crear empulaments entre les fibres dels cables que entren en aquella caixa.

La freqüència és de 25 caixes al dia. Totes les entrades i sortides són a través de la pantalla i aplicatiu client GIS. Els problemes i errors més freqüents que es troben són la falta de determinats cables o caixes per no haver-se actualitzat convenientment la xarxa.

Plotejat de rutes

Quan el *supervisor* de servei necessita esbrinar ràpidament tots els trams de canalització per on passa un determinat servei a un client el primer que fa és cercar, en un aplicatiu apart i ja en producció, per quines fibres òptiques de quin cable que arriba a la central s'està donant aquest servei.

La interfície per tant és, un cop sel·leccionat un registre final de xarxa, un diàleg en el que es mostren els cables que passen per la caixa d'empulaments final i les fibres d'aquests, de manera que es pot sel·leccionar una fibra determinada. Posteriorment la ruta completa de la fibra es ploteja sobre el plànol.

La freqüència és de 2 rutes al dia. Totes les entrades i sortides són a través de la pantalla i aplicatiu client GIS. Els problemes i errors més freqüents que es troben són la falta d'actualització de les dades de rutes.

2.6.3 Requisits d'usabilitat

El *delineant* necessita una formació aproximada de quatre hores, dues de teoria sobre el model de dades i dues de pràctica amb el client GIS. El *gestorFO* necessita una formació de dues hores, una de teoria sobre el model de dades i una de pràctica amb el client GIS. El *supervisor* necessita una hora de formació pràctica amb el client GIS.

Capítol 3

Anàlisi de requeriments

En l'anàlisi de requisits es defineixen amb detall les necessitat que ha de resoldre el programari i que han sigut declarades en el capítol anterior. Es crea una idea precisa de les funcions i requisits i s'extrauen les diferents classes i els atributs i mètodes d'aquestes.

3.1 Paquets

En aquest aplicatiu es descobreixen dos paquets de servei bàsics. El paquet *ModelDB* conté l'estructura de la base de dades que permet emmagatzemar d'una manera correcta totes les dades referents a la xarxa, tant dades geomètriques com atributs dels diferents elements que la formen. El paquet *ClientGIS* conté la funcionalitat que es pot extraure del model. La relació entre ambdós paquets és un accés a les dades des del client.

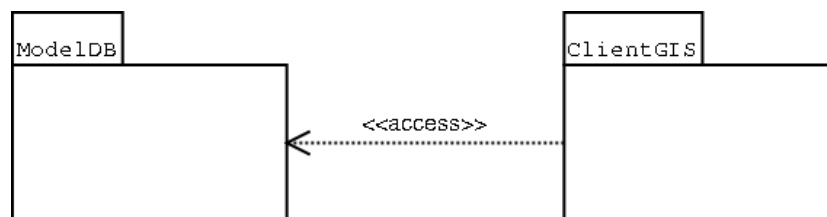


Figura 3.1: Paquets

3.2 Revisió dels casos d'ús

Els casos d'ús detectats en la fase de recollida de requeriments són suficients per la primera aproximació que es desitja en aquest treball. Tal com es comenta en la introducció s'ha limitat el domini de l'aplicatiu a desenvolupar a les funcionalitats més bàsiques i aquestes són les recollides en els casos d'ús. Els casos d'edició es podrien separar en creació, edició o modificació i eliminació, i fins i tot la modificació es podria plantejar com modificació de les dades associades o modificació de la ubicació o geometria, però aquesta dispersió no aporta molta més claredat al disseny en primera instància i sí molta més feina al multiplicar per tres el nombre de casos d'ús.

Es poden ampliar els casos d'ús amb els corresponents a detallar millor tot el que suposa per al client GIS, però no és l'abast d'aquest treball estudiar un sistema d'informació geogràfica. Igualment falten els casos d'ús de manteniment de la base de dades que suposa el model de dades de la xarxa, o els casos d'ús de gestió d'usuaris. En definitiva es troben a faltar aquells casos d'ús d'administració i manteniment del sistema.

Tenint en compte les funcionalitats que es poden arribar a extraure de la conjunció d'un model de dades d'objectes reals i un sistema GIS que situa sobre el plànol aquests objectes, potser els primers casos d'ús detectats semblen massa senzills, però la l'autolimitació és necessària per arribar a fons en aquest primer intent d'obtenció del disseny de l'aplicatiu.

En definitiva, tot i la simplicitat, són vàlids i suficients els casos d'ús detectats.

3.3 Classes d'anàlisi

Al llarg de la recollida dels requeriments es fa esmena a diferents xarxes una dins de l'altra, la de l'obra civil, la dels cables i la de les fibres. Aquesta és la idea fonamental de l'estructura de les relacions entre les classes del model de dades i cal concretar-la i definir-la perfectament, doncs a partir d'aquesta es podrà anar integrant noves capacitats futures al model i utilitzar-les des del client d'edició. Les tres xarxes que es detecten són:

- Xarxa d'obra civil: Aquest conjunt de trams i punts el formen les canalitzacions i els registres. Són les classes de les que s'edita la seva posició i traçada sobre el plànol directament amb l'eina GIS. A partir d'aquestes s'arriba a les altres classes del model. Aquesta xarxa forma un graf i es pot considerar com el primer nivell de la xarxa complerta. Els conductes o tubulars són part de les canalitzacions i permeten ordenar la disposició dels cables, que formen part del següent nivell.

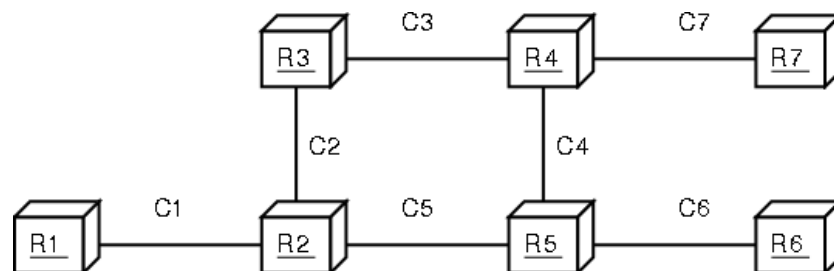


Figura 3.2: Xarxa de canalitzacions i registres

- Xarxa de cablejats: Aquest segon nivell de la xarxa està format per aquells elements portadors de fibres i que permeten la seva manipulació. Són els cables i les caixes d'empulaments. Forma un graf diferent al de primer nivell, però que hi està relacionat directament doncs tots els vèrtexs i els trams deriven del graf que forma la xarxa d'obra civil. Pot existir més d'un graf inconnex, doncs els cables no tenen perquè estar connectats entre sí.
- Xarxa de fibres: El tercer nivell forma un altre conjunt de grafs lineals que deriven del segon. Està format per les fibres i els empulaments de fibres. Cada ruta de fibres forma un graf lineal que, en el cas de connectar un client, va directament de la central al client.

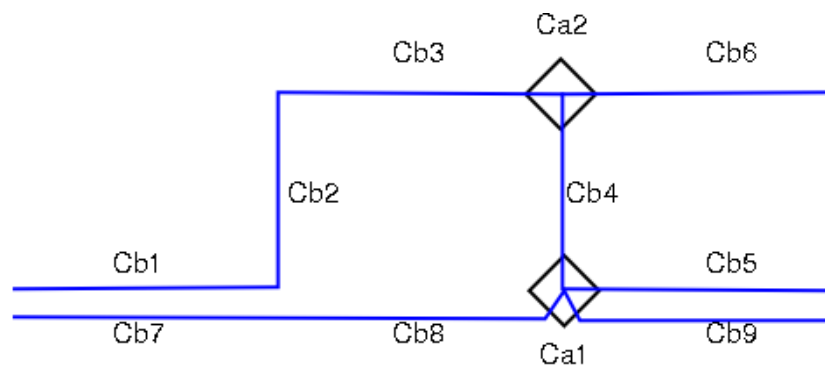


Figura 3.3: Xarxa de cables i caixes

D'altra banda en la recollida de requeriments s'ha descrit un seguit d'interfícies d'usuari que obliguen a la definició de classes de frontera per donar la funcionalitat que es necessita i que serveixen les classes de control.

En conjunt les classes es tipifiquen doncs d'acord amb la relació que defineix Campderrich [1] en classes de frontera, classes d'entitats i classes de control. Les classes de frontera i les de control es descriuen en els diagrames de col·laboració dels casos d'ús i les classes de frontera es relacionen directament amb els diagrames d'interfície d'usuari.

3.3.1 Classes d'entitat

Les classes d'entitat s'identifiquen¹ a partir dels casos d'ús.

1. Editar canalització

Classes: Canalitzacio, Traçada?, TipusCanalitzacio, Contractista, Projecte

2. Editar conducte

Classes: Conducte, Canalitzacio*, TipusConducte, Titular

3. Editar registre

Classes: Registre, Ubicacio?, Canalitzacio*, TipusRegistre, Contractista*, Projecte*

4. Editar cable

Classes: Cable, Conducte*, Canalitzacio*, Projecte*, Contractista*, Fibra

5. Editar caixa

Classes: Caixa, Registre*, Canalitzacio*, Cable*, Projecte*, Contractista*, TipusCaixa

6. Editar empiulament

Classes: Empiulament, Caixa*, Registre*, Fibra*, Cable*, Safata, Tub, Ruta, Client

7. Plotejar ruta

Classes: Traçada?*, Planol, Client*

¹Els símbols representen una classe ja identificada per '*' i una classe dubtosa per '?'

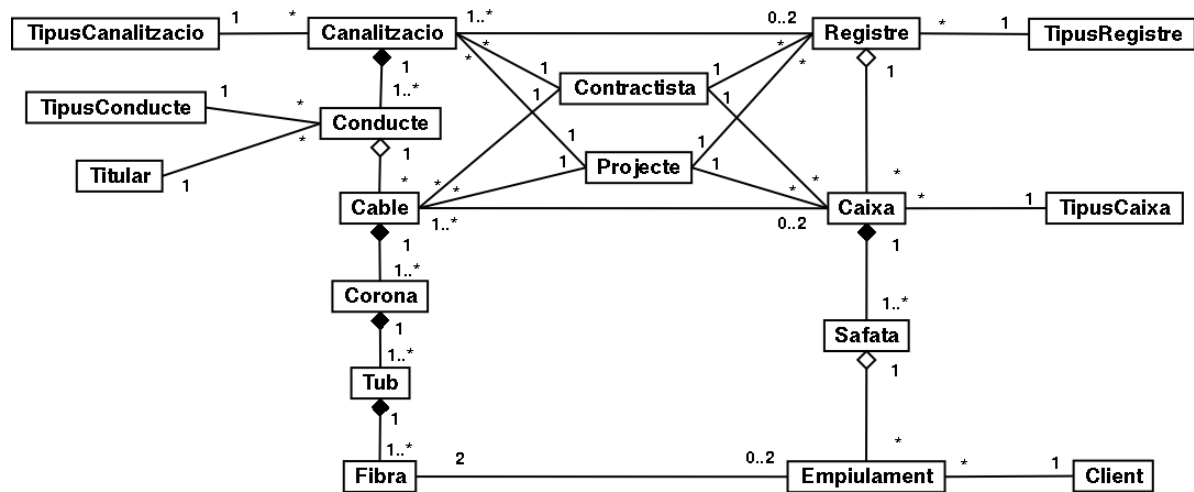


Figura 3.4: Primera revisió de les Classes

Les classes Traçada i Ubicacio són en realitat els atributs geomètrics de les classes Canalitzacio i Registre respectivament. La classe Safata correspon a un component de la classe Caixa però que té utilitat en el moment de realitzar un empiulament. Igualment la classe Tub és un component de Cable que descriu l'estructura interna del cable amb el repartiment dels tubs en corones, pot aparèixer per tant una nova classe Corona. La classe Ruta és en realitat una derivació de la traçada particular que segueix una fibra al llarg de tots els seus empiulaments per donar un servei a un client. La classe Planol correspon en realitat a un document de sortida de l'aplicatiu. Les classes TipusCanalitzacio, TipusConducte, TipusRegistre, TipusCaixa, Contractista, Projecte, Titular i Client són classes auxiliars que emmagatzemen dades complementaries.

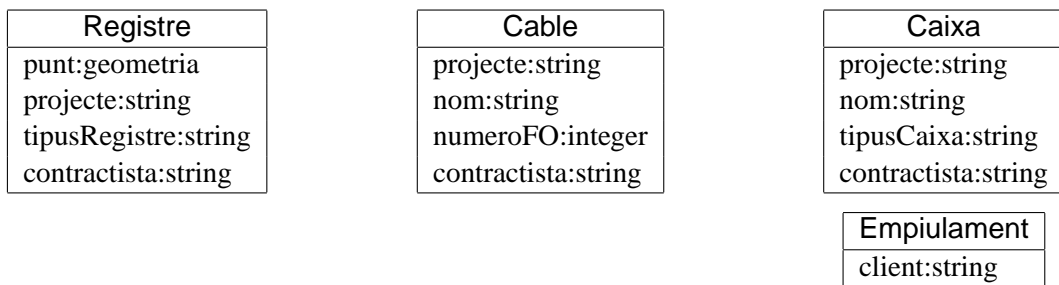
Per tant en una primera revisió es dedueix que les classes d'entitat que es troben són les enumerades a continuació i que corresponen al diagrama de la Figura 3.4.

Canalitzacio	TipusCanalitzacio	Contractista	Projecte	Conducte	TipusConducte
Titular	Registre	TipusRegistre	Cable	Fibra	Caixa
TipusCaixa	Empiulament	Safata	Tub	Corona	Client

Per fer una segona revisió s'estudia els atributs d'aquestes classes. Tal com es comprova al diagrama de classes d'entitat en primera revisió hi ha classes que poden integrar-se com atributs de les classes amb les que es relacionen; són totes aquelles classes que tenen una cardinalitat $1 \leftrightarrow *$ i que la classe de la banda amb cardinalitat 1 no té altres relacions. És el cas de TipusCanalitzacio, TipusRegistre, TipusConducte, TipusCaixa, Titular, Projecte, Contractista i Client.

Queden per tant les classes Canalitzacio, Conducte, Registre, Cable, Caixa, Empiulament, Fibra, Safata, Tub i Corona. Els atributs d'aquestes són:

Canalitzacio	Conducte
tram:geometria	tipusConducte:string
projecte:string	titular:string
tipusCanalitzacio:string	
contractista:string	



La resta de classes no tenen atributs:



3.3.2 Relacions

Relacions d'herència

La resolució dels atributs geomètrics de les dues classes que tenen 'visibilitat' des de l'aplicatiu GIS es resol amb la implementació de la interfície geomètrica de tram i de punt. És una manera de dir que els objectes de les classes **Canalitzacio** i **Registre** tenen determinades característiques geomètriques que extreuen directament de l'aplicatiu GIS gràcies a les funcionalitats que dona aquesta mena de programari.

És una qüestió que depèn molt de la solució tecnològica que s'adopti, cal resoldre-la per tant en la fase de disseny. De moment es declara la implementació de les interfícies geomètriques per part de les classes de manera que heretin aquesta capacitat gràfica.

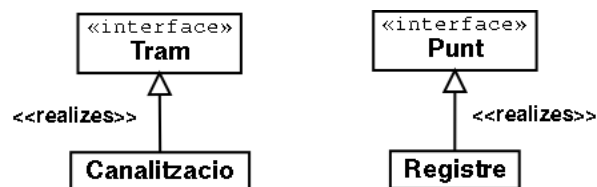


Figura 3.5: Herències

Relacions d'associació

No existeix cap classe associativa. Sobre les relacions entre classes es modifica la cardinalitat que expressa el màxim admès de nombre de fibres per cable tal com es declarava al diagrama de classes del domini (Figura 2.2), doncs és una limitació innecessària i que podria portar a un model massa restrictiu.

Les relacions existents entre les classes **Canalitzacio** i **Registre** i entre les classes **Cable** i **Caixa** segueixen la pauta detectada en la definició del domini. Cada tram de **Canalitzacio** pot relacionar-se amb cap o un **Registre** a cada un dels seus extrems i els **Registre** no poden existir sense una **Canalitzacio** que els enllaci a la xarxa i alhora poden arribar a connectar-se a través de múltiples

Canalització. Igualment cada tram de **Cable** pot tenir cap o una **Caixa** a cada un dels seus dos extrems i una **Caixa** pot connectar múltiples **Cable**.

De forma equivalent al que s'ha fet en la reducció de les classes amb cardinalitat 1, la relació existent entre les classes **Fibra** i **Empiulament** s'anul·la i la classe amb cardinalitat fixa i sense atributs **Fibra** es converteix en atribut de **Empiulament**.

Relacions d'agregació

Existeixen relacions d'agregació normals i de composició. De fet per la manera com es plantegen les diferents tasques del *delineant*, *gestorFO* i *supervisor* tota la interacció d'aquests actors amb el model es fa a través de les classes **Canalització** i **Registre**, la resta de les classes són agregades de les primeres.

En concret la classe **Conducte** compon a **Canalització** doncs els conductes formen part del prisma de la canalització. Igualment les classes **Fibra**, **Tub** i **Corona** son components indivisibles de **Cable** doncs defineixen l'estructura interna d'un cable. La classe **Safata** també forma part de **Caixa**.

D'altra banda les relacions entre **Empiulament** i **Safata**, entre **Caixa** i **Registre** i entre **Cable** i **Conducte** són agregacions normals amb un significat de *continent* ↔ *contingut* on les primeres s'ubiquen dins de les segones.

Les classes que no tenen atributs i una relació de composició amb cardinalitat evidentment fixa en la seva banda de la relació com es el cas de **Corona**, **Tub**, i **Safata** es redueixen a atributs de la classe **Empiulament**.

Empiulament
client:string
corona1:integer
tub1:integer
fibra1:integer
corona2:integer
tub2:integer
fibra2:integer
safata:integer

Resultat de les relacions

Queda finalment el diagrama de classes en segona revisió (Figura 3.6). Es redueixen les classes segons les explicacions donades en els apartats anteriors.

3.3.3 Diagrames de col·laboració

Es descriu un diagrama de col·laboració simplificat per cada cas d'ús. En aquests diagrames es fa servir la notació estàndard UML [2] per les classes de frontera, de control i d'entitat ja esmentades.

1. Editar canalització

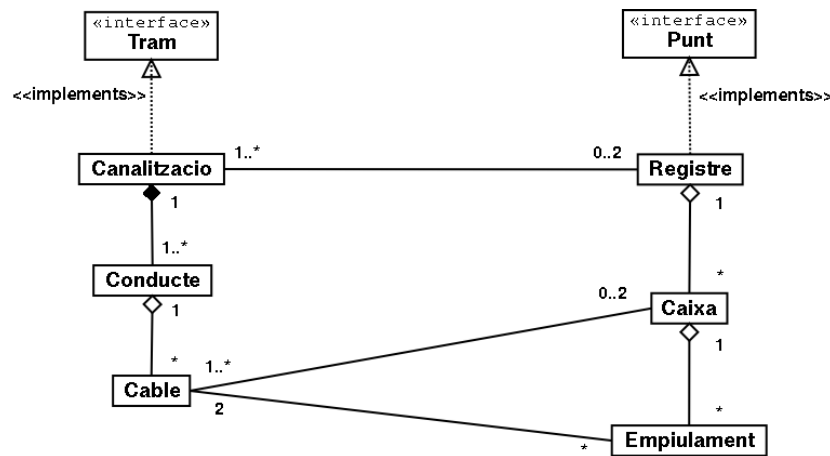


Figura 3.6: Segona revisió de les Classes

El *delineant* davant de l'aplicatiu GIS pot crear una nova traça que en dibuixar-la queda sel·leccionada automàticament i es convertirà en un objecte de **Canaltzació** al donar-li les dades a través del diàleg corresponent, o sel·leccionar un tram de canaltzació per modificar-lo, tant en les dades associades a través del diàleg com en la geometria directament a través del programari GIS. Igualment es pot sel·leccionar per eliminar-lo. A la Figura 3.7 hi ha el diagrama de col·laboració simplificat.

2. Editar conducte

El *delineant* davant de l'aplicatiu GIS sel·lecciona el tram de canaltzació del que vol editar els conductes. Un cop marcat el tram per l'aplicatiu pot editar les dades dels conductes a través del diàleg corresponent. A la Figura 3.8 hi ha el diagrama de col·laboració simplificat.

3. Editar registre

El *delineant* davant de l'aplicatiu GIS pot crear un nou registre que en dibuixar-lo queda sel·leccionat automàticament i es convertirà en un objecte de **Registre** al donar-li les dades a través del diàleg corresponent, o sel·leccionar un registre per modificar-lo, tant en les dades associades a través del diàleg com en la geometria directament a través del programari GIS. Igualment es pot sel·leccionar per eliminar-lo. A la Figura 3.9 hi ha el diagrama de col·laboració simplificat.

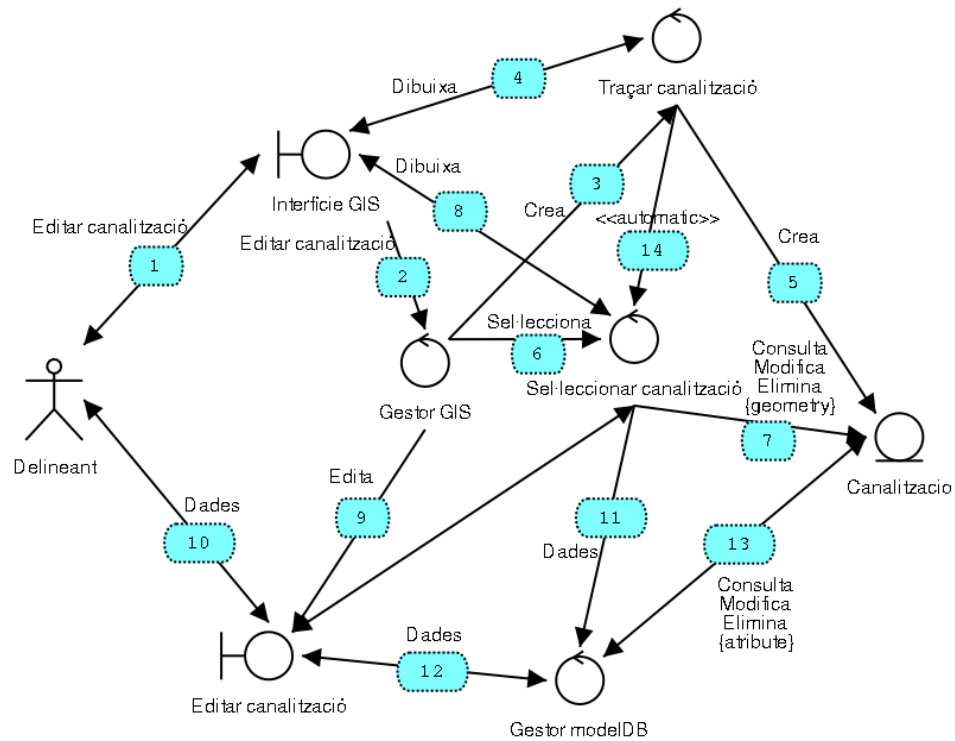
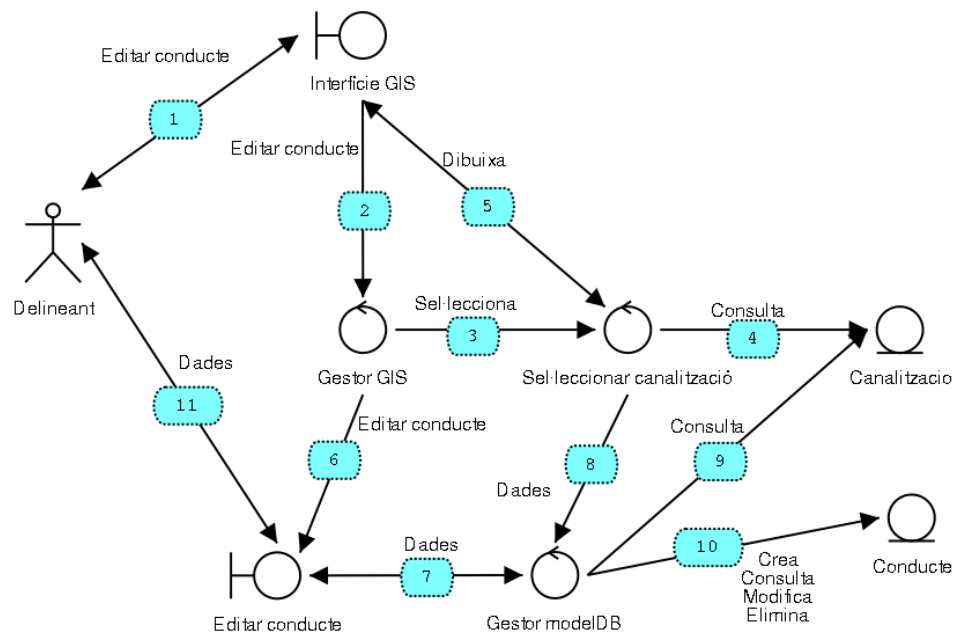
4. Editar cable

El *delineant* davant de l'aplicatiu GIS sel·lecciona el tram de canaltzació del que vol editar els cables que hi passen. Un cop marcat el tram per l'aplicatiu pot editar les dades dels cables a través del diàleg corresponent. A la Figura 3.10 hi ha el diagrama de col·laboració simplificat.

5. Editar caixa

El *delineant* davant de l'aplicatiu GIS sel·lecciona el registre del que vol editar les caixes que conté. Un cop marcat el punt per l'aplicatiu pot editar les dades de les caixes a través del diàleg corresponent. A la Figura 3.11 hi ha el diagrama de col·laboració simplificat.

6. Editar empiulament

Figura 3.7: Diagrama de col·laboració d'**Editar canalització**Figura 3.8: Diagrama de col·laboració d'**Editar conducte**

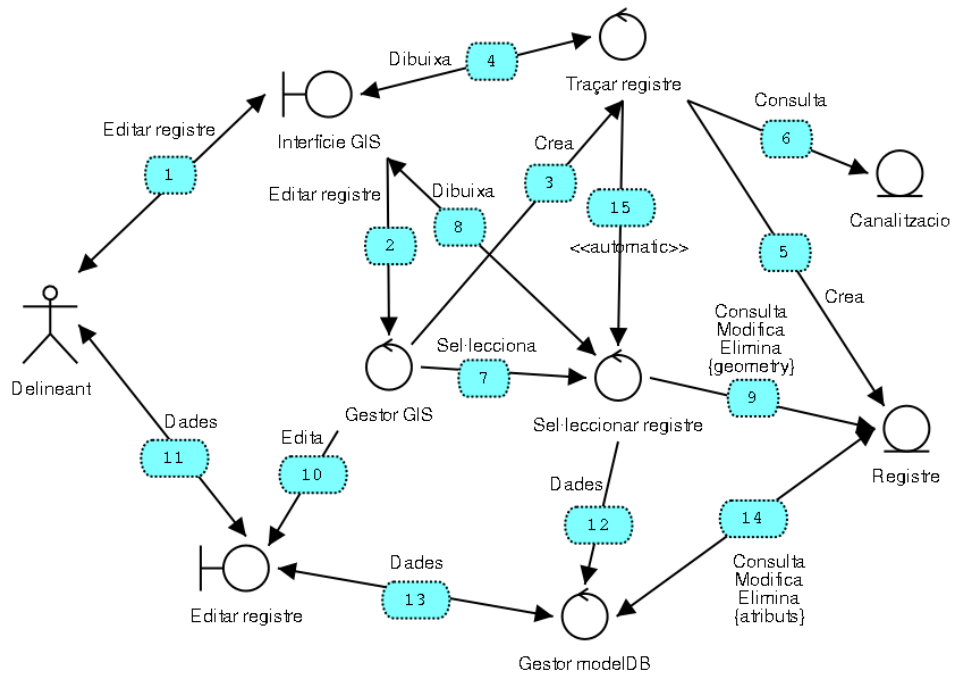


Figura 3.9: Diagrama de col·laboració d'Editar registre

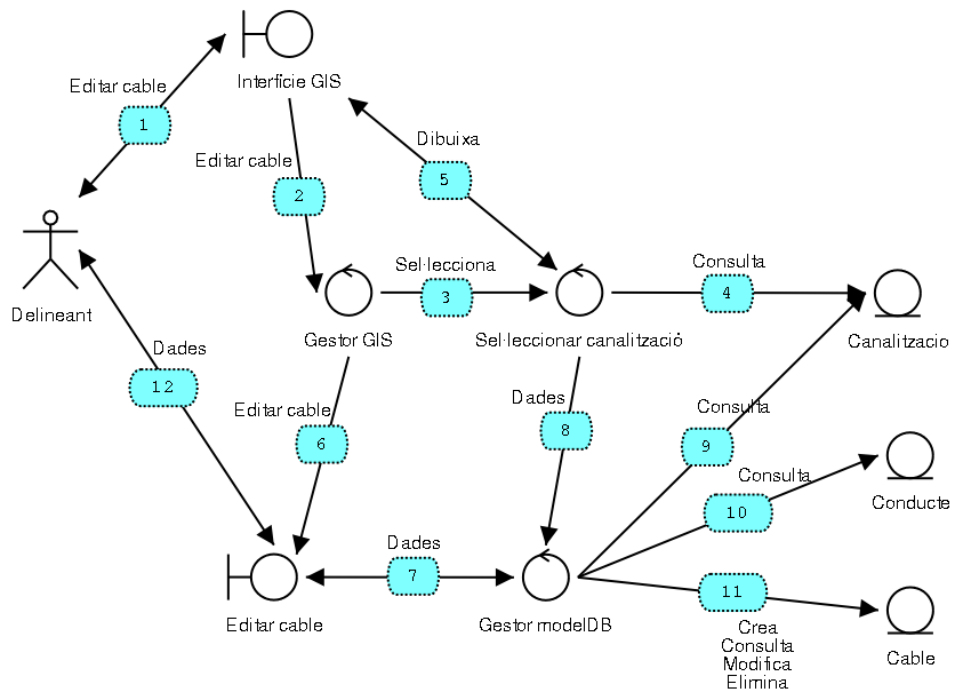


Figura 3.10: Diagrama de col·laboració d'Editar cable

El *gestorFO* davant de l'aplicatiu GIS sel·lecciona el registre del que vol editar els empiulaments de les caixes que conté. Un cop marcat el punt per l'aplicatiu pot editar els empiulaments a través del diàleg corresponent. A la Figura 3.12 hi ha el diagrama de col·laboració simplificat.

7. Plotejar ruta

El *supervisor* davant de l'aplicatiu GIS sel·lecciona el registre del que vol sel·leccionar la fibra dins de les caixes que conté. Un cop marcat el punt per l'aplicatiu pot sel·leccionar la fibra a través del diàleg corresponent i ordenar que es dibuixi en el plànol la ruta completa. A la Figura 3.13 hi ha el diagrama de col·laboració simplificat.

3.3.4 Diagrama estàtic d'anàlisi

El diagrama estàtic d'anàlisi (Figura 3.14) recull totes les classes de frontera, control i entitat detectades. Inclou els atributs i mètodes de cada classe i les diferents relacions entre les classes.

3.4 Especificació formal dels casos d'ús

Es descriuen amb detall les seqüències d'operacions i missatges entre les diferents classes d'entitat, frontera i control detectades en els diagrames de col·laboració dels casos d'ús.

1. Editar canalització

- (a) El *delineant* accedeix a l'aplicatiu GIS per editar una canalització (1 i 2).
- (b) Si desitja crear-la ho expressa al gestor del GIS a través de la comanda corresponent, normalment un botó (3).
- (c) A través de la interfície traça sobre el plànol la canalització (4).
- (d) Un cop dibuixada la canalització es crea com **Canalitzacio** en la base de dades (5). També queda sel·leccionada automàticament pel propi aplicatiu GIS (14).
- (e) Si el *delineant* desitja modificar una canalització existent primer cal sel·leccionar-la i ho expressa al gestor a través de la comanda corresponent, normalment un botó (6).
- (f) La sel·lecció es realitza a través de la interfície GIS que accedeix a les dades del tram en el model de dades. Igualment es pot modificar a través de la interfície GIS la geometria de la traçada (7 i 8).
- (g) Un cop sel·leccionada la canalització s'accedeix a les seves dades a través del diàleg corresponent d'editar la canalització (9 i 10).
- (h) El diàleg rep l'identificador de la canalització de la que es tracta com un codi des del sistema GIS (11) a través del gestor de dades (12).
- (i) El *delineant* actualitza les dades associades a la canalització amb el diàleg, que es connecta a través del gestor de dades a l'objecte de **Canalitzacio**(13).

2. Editar conducte

- (a) El *delineant* accedeix a l'aplicatiu GIS per editar els conductes d'una canalització (1 i 2).

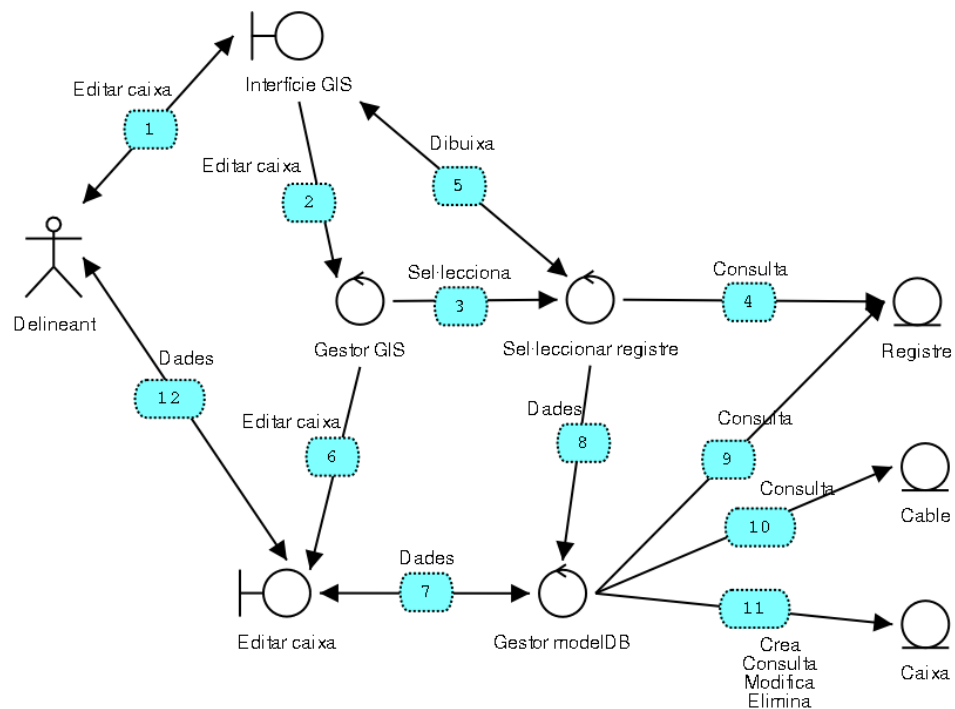


Figura 3.11: Diagrama de col·laboració d'**Editar caixa**

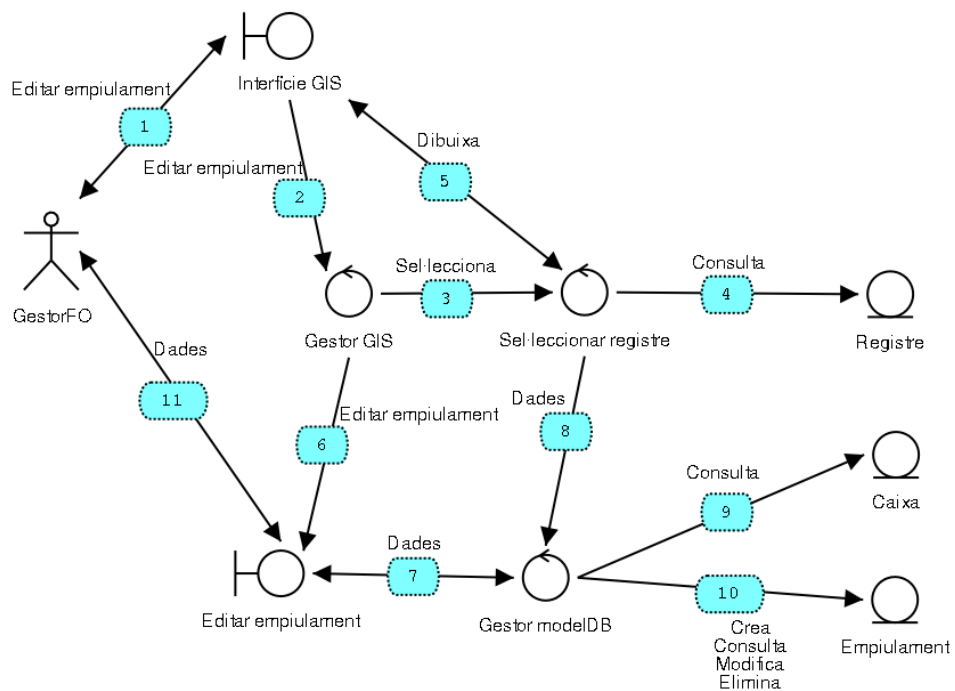


Figura 3.12: Diagrama de col·laboració d'**Editar empiulament**

- (b) Primer cal sel·leccionar de quina canalització es tracta, el que es realitza igual que en el cas d'ús **Editar canalització**; es comanda al gestor la sel·lecció d'un tram de canalització normalment a través d'un botó (3) i se sel·lecciona a la interfície GIS (4 i 5).
- (c) Un cop sel·leccionada la canalització s'accedeix al diàleg d'edició dels tubulars amb una comanda del gestor GIS (6).
- (d) El diàleg rep l'identificador de la canalització de la que es tracta com un codi des del sistema GIS (8) a través del gestor de dades (7).
- (e) El *delineant* actualitza les dades dels conductes components de la canalització a través del diàleg d'edició (11), que es connecta a les dades a través del gestor (9 i 10).

3. Editar registre

- (a) El *delineant* accedeix a l'aplicatiu GIS per editar un registre (1 i 2).
- (b) Si desitja crear-lo ho expressa al gestor del GIS a través de la comanda corresponent, normalment un botó (3).
- (c) A través de la interfície traça sobre el plànol el registre (4). L'aplicatiu GIS comprova que el registre es situa en l'extrem d'una canalització (6).
- (d) Un cop dibuixat el registre es crea com **Registre** en la base de dades (5). També queda sel·leccionat automàticament pel propi aplicatiu GIS (15).
- (e) Si el *delineant* desitja modificar un registre existent primer cal sel·leccionar-lo i ho expressa al gestor a través de la comanda corresponent, normalment un botó (7).
- (f) La sel·lecció es realitza a través de la interfície GIS que accedeix a les dades del registre en el model de dades. Igualment es pot modificar a través de la interfície GIS la geometria de la posició (8 i 9).
- (g) Un cop sel·leccionat el registre s'accedeix a les seves dades a través del diàleg corresponent d'editar el registre (10 i 11).
- (h) El diàleg rep l'identificador del registre del que es tracta com un codi des del sistema GIS (12) a través del gestor de dades (13).
- (i) El *delineant* actualitza les dades associades al registre amb el diàleg, que es connecta a través del gestor de dades a l'objecte de **Registre**(14).

4. Editar cable

- (a) El *delineant* accedeix a l'aplicatiu GIS per editar els cables que recorren pels conductes d'una canalització (1 i 2).
- (b) Primer cal sel·leccionar de quina canalització es tracta, el que es realitza igual que en el cas d'ús **Editar canalització**; es comanda al gestor la sel·lecció d'un tram de canalització normalment a través d'un botó (3) i se sel·lecciona a la interfície GIS (4 i 5).
- (c) Un cop sel·leccionada la canalització s'accedeix al diàleg d'edició dels cables amb una comanda del gestor GIS (6).
- (d) El diàleg rep l'identificador de la canalització de la que es tracta com un codi des del sistema GIS (8) a través del gestor de dades (7).
- (e) El *delineant* actualitza les dades dels cables que passen pels conductes de la canalització a través del diàleg d'edició (12), que es connecta a les dades a través del gestor (9, 10 i 11).

5. Editar caixa

- (a) El *delineant* accedeix a l'aplicatiu GIS per editar les caixes d'un registre (1 i 2).
- (b) Primer cal sel·leccionar de quin registre es tracta, el que es realitza igual que en el cas d'ús **Editar registre**; es comanda al gestor la sel·lecció d'un registre normalment a través d'un botó (3) i se sel·lecciona a la interfície GIS (4 i 5).
- (c) Un cop sel·leccionat el registre s'accedeix al diàleg d'edició de les caixes amb una comanda del gestor GIS (6).
- (d) El diàleg rep l'identificador del registre del que es tracta com un codi des del sistema GIS (8) a través del gestor de dades (7).
- (e) El *delineant* actualitza les dades de les caixes contingudes en el registre a través del diàleg d'edició (12), que es connecta a les dades a través del gestor (9, 10 i 11).

6. Editar empiulament

- (a) El *gestorFO* accedeix a l'aplicatiu GIS per editar els empiulaments d'una caixa col·locada en un registre (1 i 2).
- (b) Primer cal sel·leccionar de quin registre es tracta, el que es realitza igual que en el cas d'ús **Editar registre**; es comanda al gestor la sel·lecció d'un registre normalment a través d'un botó (3) i se sel·lecciona a la interfície GIS (4 i 5).
- (c) Un cop sel·leccionat el registre s'accedeix al diàleg d'edició dels empiulaments amb una comanda del gestor GIS (6).
- (d) El diàleg rep l'identificador del registre del que es tracta com un codi des del sistema GIS (8) a través del gestor de dades (7).
- (e) El *gestorFO* actualitza les dades dels empiulaments de les caixes contingudes en el registre a través del diàleg d'edició (11), que es connecta a les dades a través del gestor (9 i 10).

7. Plotejar ruta

- (a) El *supervisor* accedeix a l'aplicatiu GIS per descobrir la traçada total d'una ruta que connecta un client amb fibra òptica d'un cable determinat que passa per la caixa d'un registre (1 i 2).
- (b) Primer cal sel·leccionar de quin registre es tracta, el que es realitza igual que en el cas d'ús **Editar registre**; es comanda al gestor la sel·lecció d'un registre normalment a través d'un botó (3) i se sel·lecciona a la interfície GIS (4 i 5).
- (c) Un cop sel·leccionat el registre s'accedeix al diàleg de sel·lecció de la fibra òptica determinada per on passa el servei amb una comanda del gestor GIS (6).
- (d) El diàleg rep l'identificador del registre del que es tracta com un codi des del sistema GIS (8) a través del gestor de dades (7).
- (e) El *supervisor* sel·lecciona la fibra a partir de les caixes que hi ha dins del registre (9, 10 i 11).
- (f) Finalment es ploteja la ruta sobre el plànol en la pantalla de la interfície GIS (12).

3.4.1 Diagrames de seqüències

En els diagrames de seqüències de la Figura 3.15 a la Figura 3.21 s'exposa gràficament la descripció detallada de l'apartat anterior.

3.4.2 Diagrames d'estats

Els objectes del model no presenten en aquesta primera versió variacions de comportament al llarg del temps; per tant no és necessària la descripció dels estats de les diferents classes descrites aquesta anàlisi i no existeix un diagrama d'estats per cap de les classes. L'única complicació sorgeix dels diferents estats en que es poden trobar els elements gràfics en el programari GIS i la seva relació amb el model de dades, però la seva anàlisi correspon més al propi programari GIS que no pas a aquest treball.

En un futur es poden considerar estats dels elements de la xarxa que porten a diferents reaccions per part del sistema; entre aquests estats es poden considerar l'estat *avariat* quan un element es trenca o pateix una avaria. Igualment es poden considerar els diferents estats pels que passa cada un dels elements de la xarxa des de la seva concepció fins la seva eliminació, aquests serien per exemple els representats a la Figura 3.22.

3.5 Interfície d'usuari

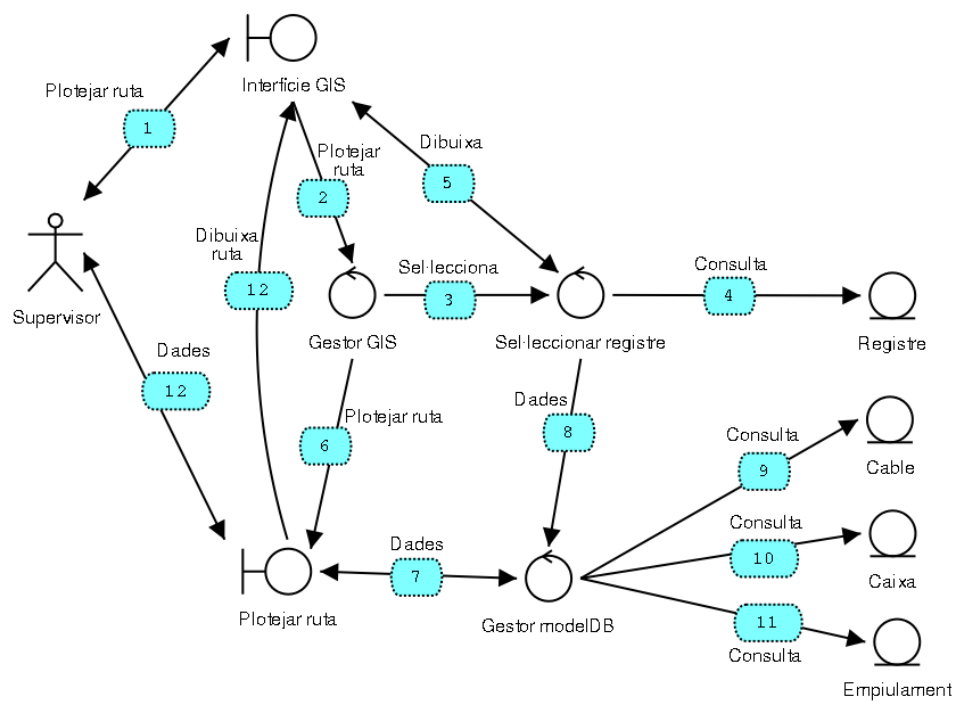
La interfície d'usuari respon a les classes frontera detectades en els diagrames de col·laboració dels casos d'ús. Cada interfície concreta les diferents operacions que realitza l'actor corresponent en la seva interacció amb l'aplicatiu.

De la Figura 3.23 a la Figura 3.29 es representen una primera aproximació de com és cada una d'aquestes interfícies. Totes corresponen a diàleg d'interacció entre l'aplicatiu i l'actor.

3.6 Revisió dels paquets

La separació en dos paquets expressada anteriorment (Figura 3.1) segueix vigent, però en les diferents seccions d'aquesta anàlisi han aparegut classes que no corresponen exactament a cap d'aquests paquets. En concret es detecten classes que no corresponen al model de dades ni al client GIS sinó directament al programari GIS. Cal fer aquesta distinció doncs les classes d'aquest paquet *AplicatiuGIS* no es desenvolupen en aquest treball sinó que són part dels programaris GIS estàndard.

Les classes que formen part del paquet *AplicatiuGIS* són InterficieGIS, GestorGIS, TraçarCanalitzacio, TraçarRegistre, Sel-leccionarCanalitzacio i Sel-leccionarRegistre. A la Figura 3.30 es representa aquesta nova disposició dels paquets.

Figura 3.13: Diagrama de col·laboració de **Plotejar ruta**

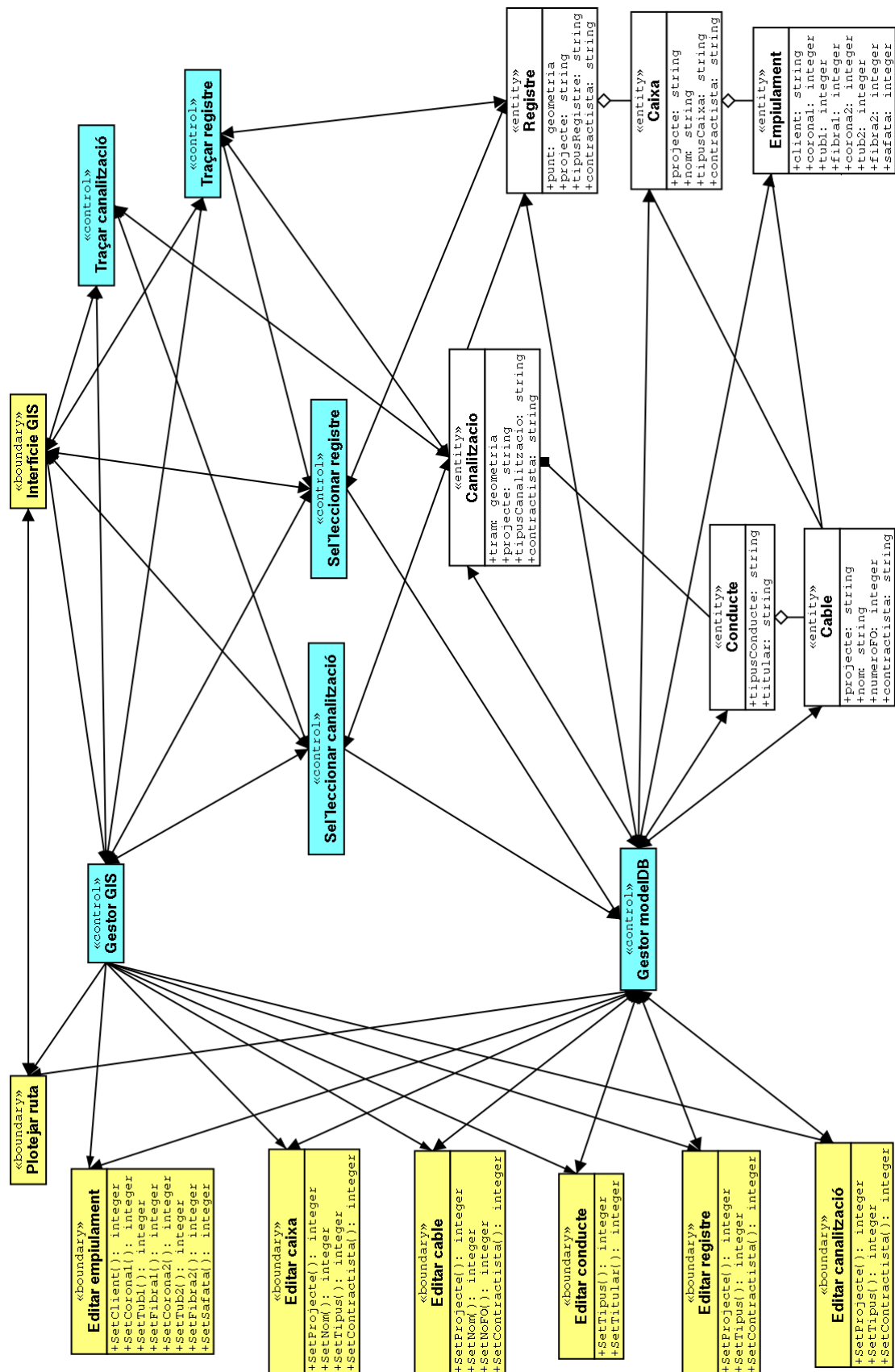


Figura 3.14: Diagrama estàtic d'anàlisi

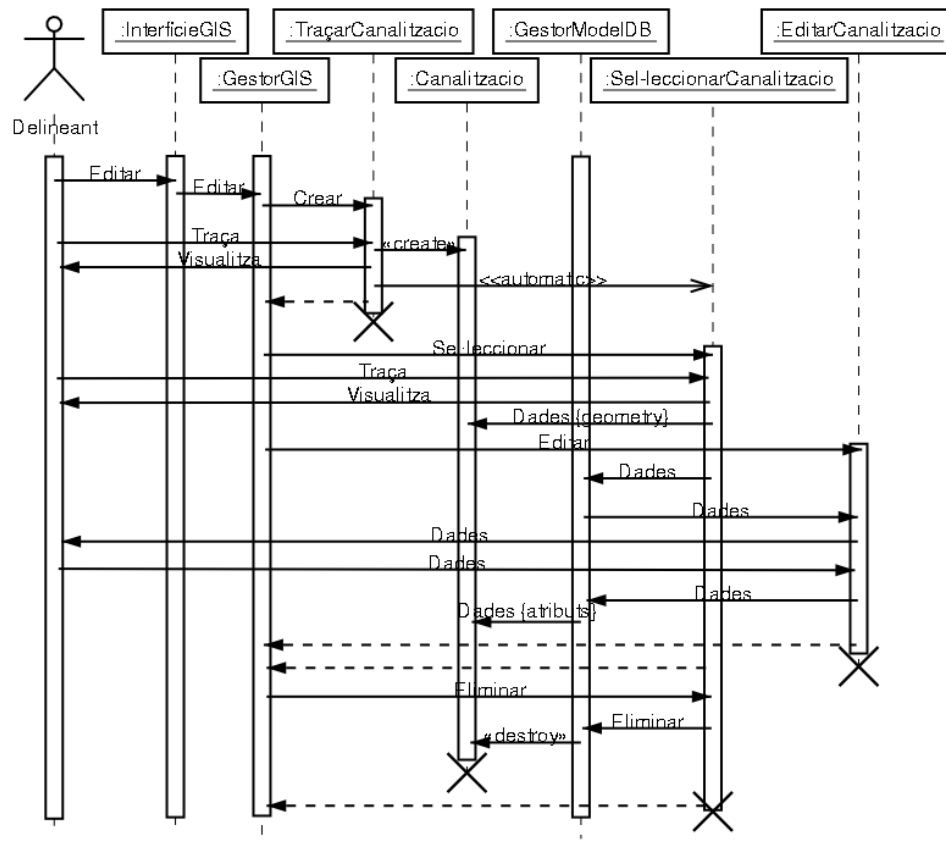


Figura 3.15: Diagrama de seqüències d'**Editar canalització**

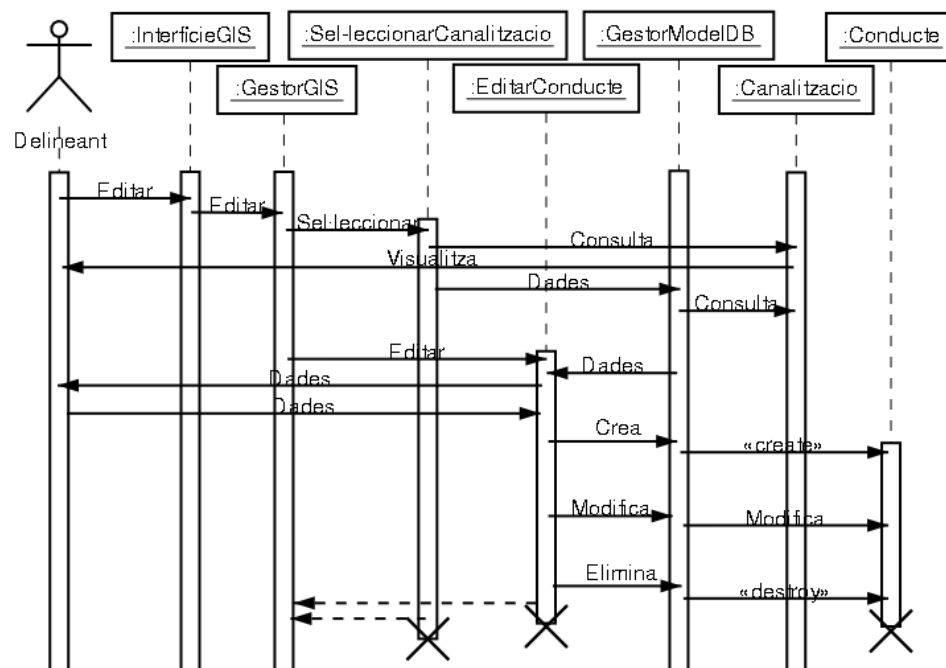


Figura 3.16: Diagrama de seqüències d'**Editar conducte**

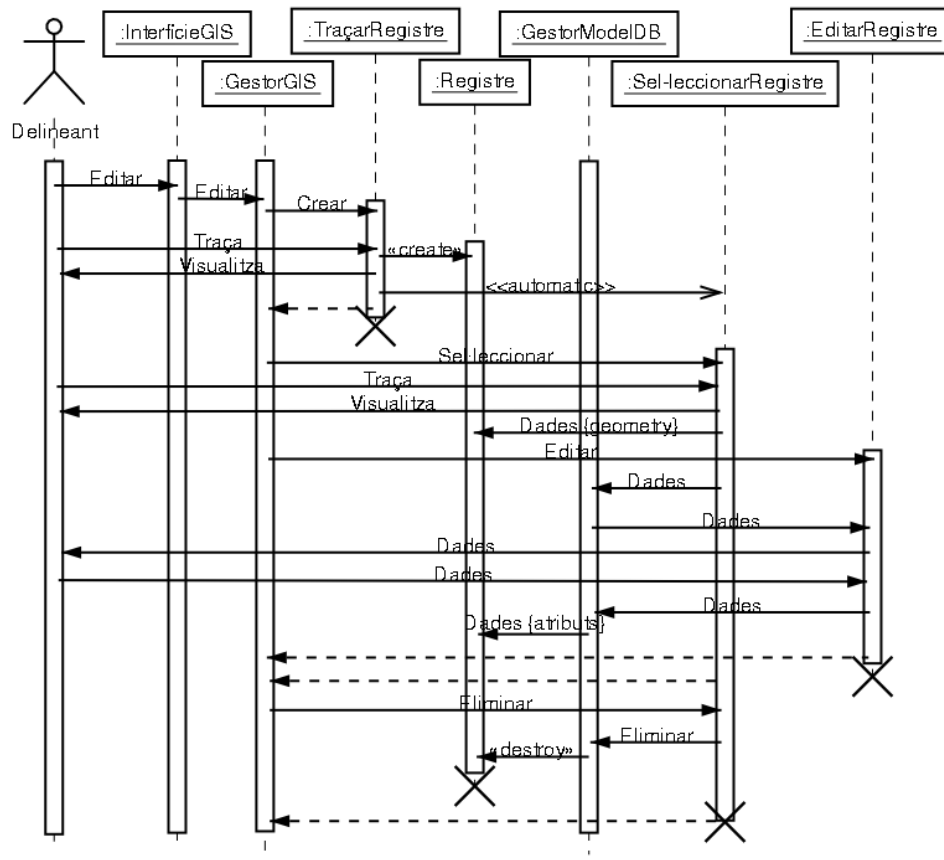


Figura 3.17: Diagrama de seqüències d'Editar registre

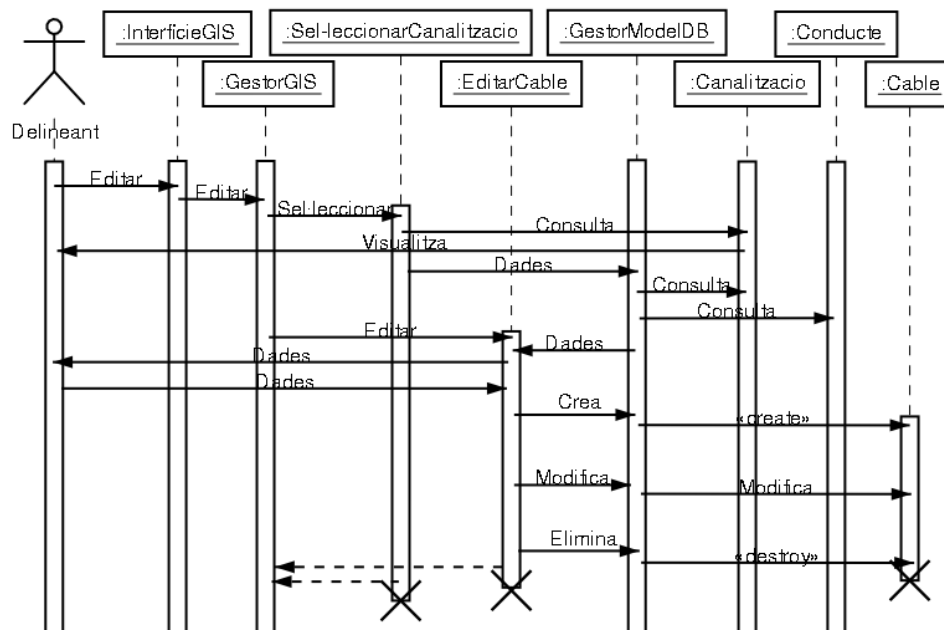
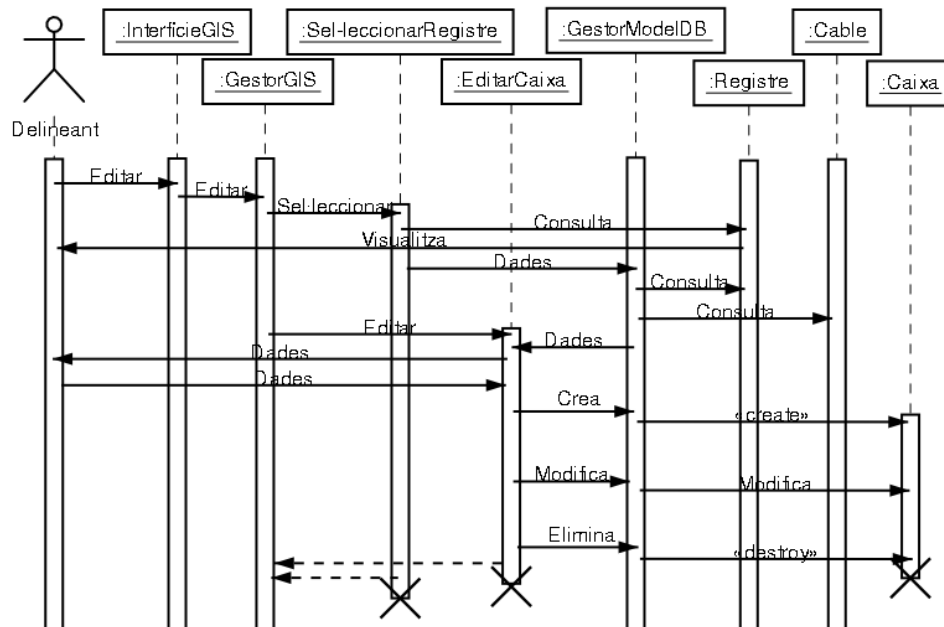
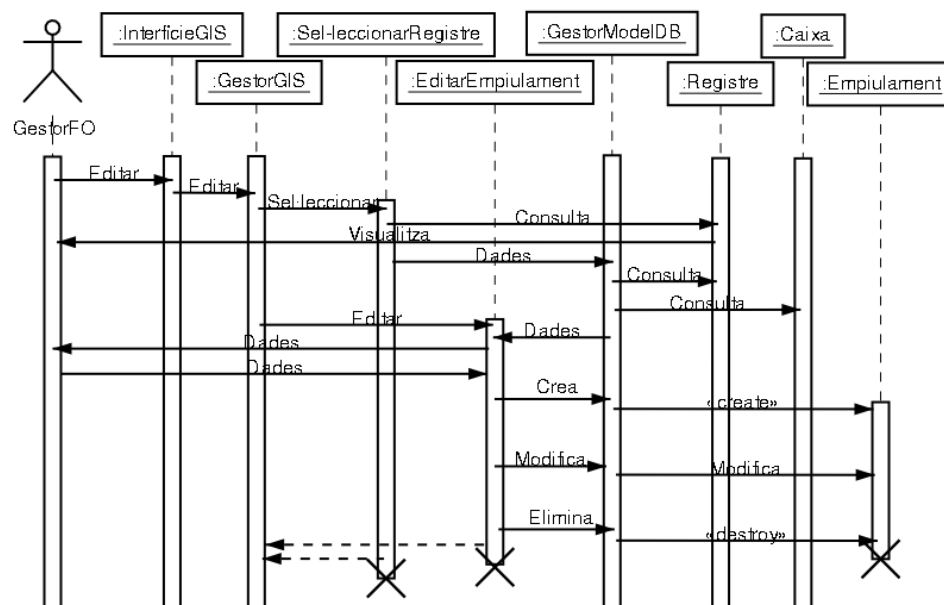


Figura 3.18: Diagrama de seqüències d'Editar cable

Figura 3.19: Diagrama de seqüències d'**Editar caixa**Figura 3.20: Diagrama de seqüències d'**Editar empiulament**

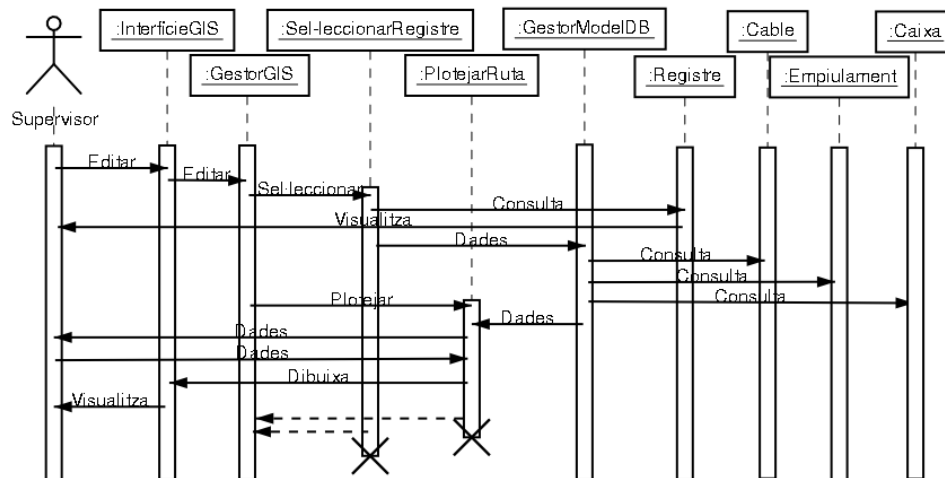


Figura 3.21: Diagrama de seqüències de **Plotejar ruta**

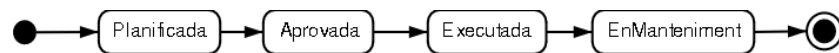


Figura 3.22: Diagrama d'Estats



Figura 3.23: Interfície de la classe frontera **EditarCanalitzacio**

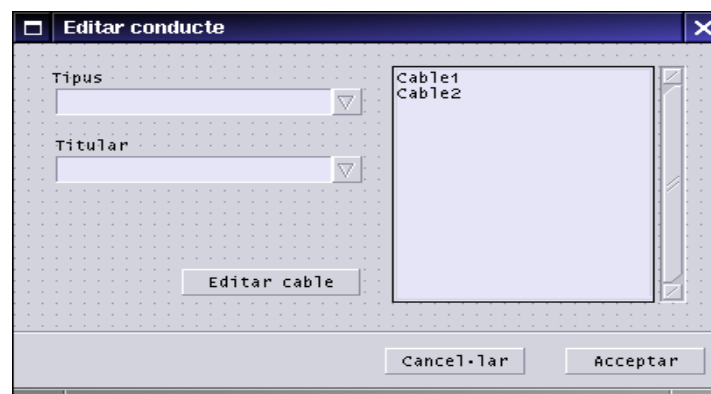


Figura 3.24: Interfície de la classe frontera **EditarConducte**

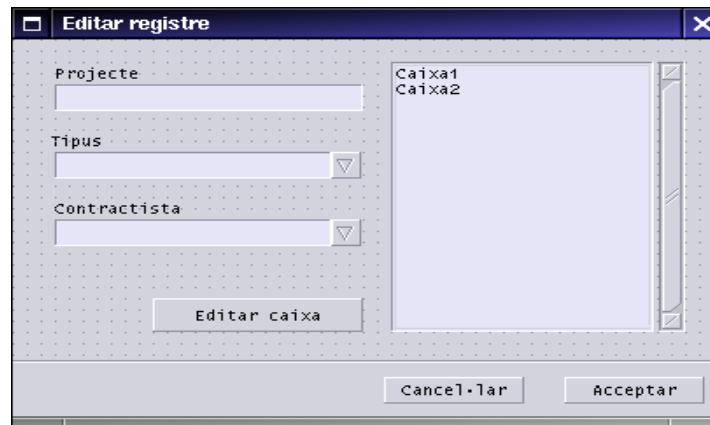


Figura 3.25: Interfície de la classe frontera EditarRegistre

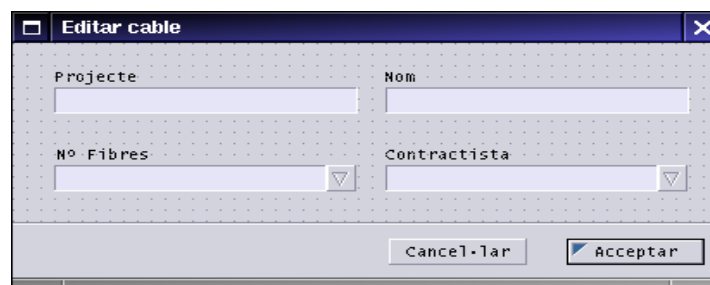


Figura 3.26: Interfície de la classe frontera EditarCable

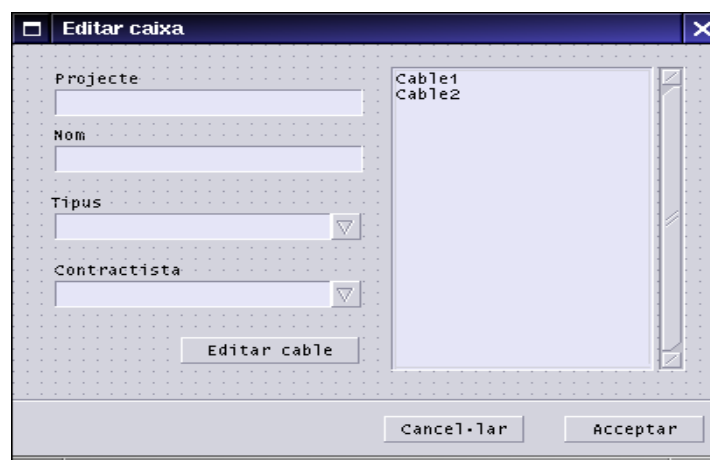


Figura 3.27: Interfície de la classe frontera EditarCaixa

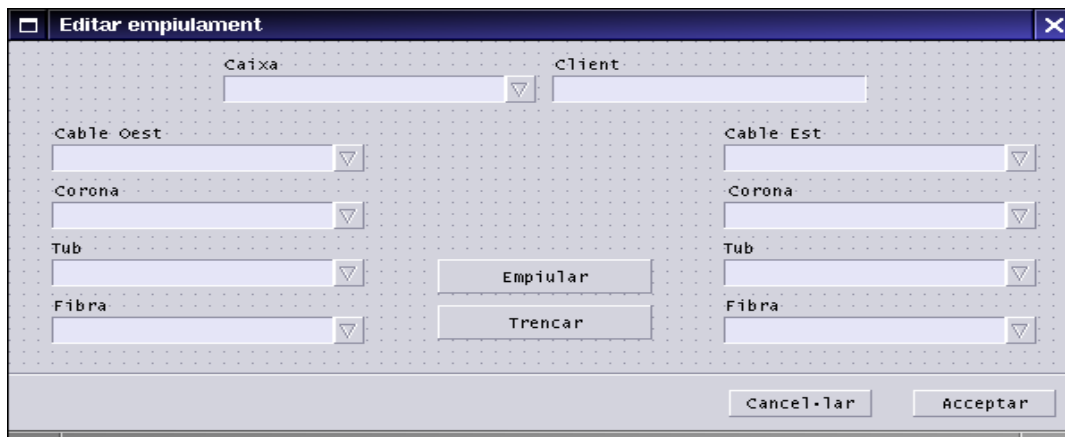


Figura 3.28: Interfície de la classe frontera EditarEmpiulament



Figura 3.29: Interfície de la classe frontera PlotejarRuta

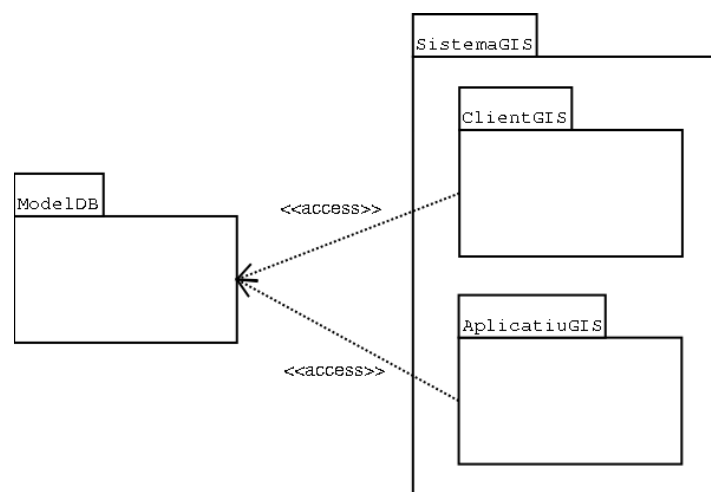


Figura 3.30: Revisió dels paquets

Capítol 4

Disseny

Un cop realitzada la recollida de requeriments i l'anàlisi d'aquests cal adaptar aquesta anàlisi a la implementació amb el disseny de la solució. S'adapta aquest disseny als aplicatius escollits que es descriuen al Capítol 9.

4.1 Arquitectura

A partir de l'anàlisi de paquets (Figura 3.1) i de les diferents pautes exposades en els capítols anteriors es descobreixen dos paquets a desenvolupar:

- D'una banda el repositori de les dades en forma d'un model de dades que utilitza un sistema gestor de base de dades DBMS (DataBase Management System).
- D'altra banda un client dins del programari GIS (Geographic Information System) que permet extraure la funcionalitat del model utilitzant una interfície adequada.

El primer paquet fa servir un subsistema gestor de bases de dades relacionals RDBMS de codi obert [8] i solvència demostrada que permet fins i tot la utilització de llenguatges procedimentals. Es tracta de **PostgreSQL** [20]. Les extensions que mantenen la coherència de la xarxa i que permeten extraure la informació desitjada com és el cas de les rutes (cas d'ús **Plotejar ruta**) es desenvolupen com funcions pròpies del RDBMS amb el llenguatge **PL/pgSQL**, molt semblant al conegut **PL/SQL** de l'empresa **Oracle** [16]. D'aquesta manera no es carrega tant el codi del client de GIS i es fa servir directament el motor del RDBMS.

La geometria de les classes que necessiten atributs geomètrics (Figura 3.5) **Canalitzacio i Registre** s'aconsegueix amb una extensió del RDBMS que atorga funcionalitat geomètrica a les taules que ho necessiten. Es tracta de **PostGIS** [22], programari de codi obert que permet convertir **PostgreSQL** en un servidor de dades espacials a l'estil del SDE (Spatial Data Engine) de l'empresa **ESRI** [17] o les extensions espacials d'**Oracle**.

El segon paquet fa servir un subsistema GIS de codi obert escrit completament en **Java**TM i que permet afegir fàcilment extensions. Es tracta de **JUMP** [21]. El client que es desenvolupa en aquest treball sobre aquest aplicatiu es programa també en **Java**TM.

Entre ambdós paquets la connexió està garantida gràcies a la utilització d'una extensió de **JUMP** que permet connectar aquest GIS amb bases de dades geomètriques sobre **PostgreSQL**. Es tracta de **PostGIS JUMP Driver** [23], un plug-in de codi obert que simplifica la utilització de **JDBCTM**. Aquesta connexió es fa a través de la xarxa localhost, intranet o internet entre les interfícies de connexió **JDBCTM** i els serveis de servidor (normalment a través del port 5432) de **PostgreSQL**.

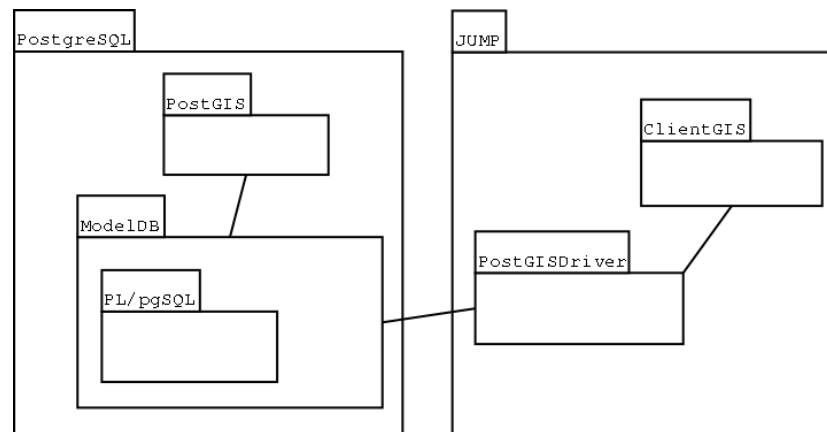


Figura 4.1: Diagrama dels subsistemes de l'arquitectura

En definitiva aquest treball tracta dels dos subsistemes *ModelDB* i *ClientGIS*, que estan relacionats amb la resta de subsistemes existents segons el diagrama de la Figura 4.1.

4.1.1 Configuració de la xarxa

La connexió entre ambdós subsistemes es realitza amb un model de client/servidor. El client és el programari GIS amb *ClientGIS* i el servidor és el RDBMS que ofereix dades i altres funcionalitats associades amb les dades.

No és la intenció d'aquest treball desenvolupar el programari de connexió entre el client i el servidor, doncs es fan servir subsistemes que ja donen aquesta funcionalitat de comunicació.

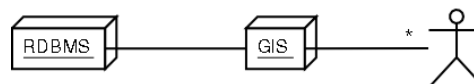


Figura 4.2: Nodes de comunicació de l'arquitectura

4.1.2 Punt de partida dels subsistemes

L'estat inicial dels subsistemes detectats (Figura 4.1) en el moment d'iniciar el programari és:

- Per la banda del RDBMS *PostgreSQL* aquest està en marxa amb la base de dades carregada. Aquesta base de dades que conté l'esquema del model *ModelDB* també s'ha extés amb

PostGIS. Igualment s'han carregat els mòduls *PL/pgSQL*. La política d'usuaris és correcta i el servidor de dades actua en una màquina amb adreça i port coneguts.

- Per la banda del GIS *JUMP* aquest està en marxa amb l'extensió que permet la connexió al RDBMS carregada *PostGISDriver*. També s'ha carregat el mòdul *ClientGIS* com una extensió o com un plug-in.

4.2 Casos d'Ús

Cas d'ús **Editar canalització**:

1. El *delineant* defineix la canalització en la que desitja treballar.
 - (a) Traça una nova canalització en la interfície GIS (espai de dibuix de *JUMP*). Un cop traçada aquesta queda sel·leccionada per defecte.
 - (b) Altrament sel·lecciona la canalització directament sobre la interfície GIS (espai de dibuix de *JUMP*).
2. Escull la comanda `EditarCanalitzacio` de `GestorGIS`, accessible a través del menú o d'un botó de la barra d'eines incorporada a la interfície GIS. Això crea una instància de la classe `DialegEditarCanalitzacio`. Aquest diàleg accedeix a la funció `ConductesDeCanalitzacio` del `GestorModelDB` per mostrar els conductes relacionats amb aquesta canalització si existeixen i si existeix la `Canalitzacio`.
3. S'introdueixen o modifiquen les dades associades a la canalització sel·leccionada. Es poden escollir les opcions:
 - (a) Acceptar. De manera que es crida la funció `ModificarCanalitzacio` de `DialegEditarCanalitzacio` que crida a `EditarCanalitzacio` de `GestorModelDB` amb les dades corresponents. Si la canalització en concret no existeix encara es crea amb `CrearCanalitzacio` de `GestorModelDB`. Després es tanca el diàleg.
 - (b) Cancel·lar. Tanca directament el diàleg.
 - (c) Sel·leccionar un conducte de la llista i pitjar el botó per editar-lo. Es crida la funció `EditarConducte` de `GestorGIS` amb el conducte sel·leccionat com paràmetre.
 - (d) Sel·leccionar un conducte de la llista i pitjar el botó per eliminar-lo. Es crida la funció `EliminarConducte` de `GestorModelDB` amb el conducte sel·leccionat com paràmetre.
 - (e) Pitjar el botó per crear un nou conducte. Es crida la funció `CrearConducte` de `GestorModelDB`. Posteriorment es torna a llegir la funció `ConductesDeCanalitzacio` del `GestorModelDB` per mostrar els conductes relacionats amb aquesta canalització. Es crida finalment a `EditarConducte` de `GestorGIS` amb el conducte creat com paràmetre.

Cas d'ús **Editar conducte**

1. El *delineant* sel·lecciona la canalització de la que vol editar els conductes. La sel·lecciona directament sobre la interfície GIS, l'espai de dibuix de *JUMP*. Segueix el cas d'ús **Editar canalització** fins arribar al punt en que es pitja el botó per editar el conducte sel·leccionat de la llista o per crear un nou conducte.
2. Es crea una instància de la classe `DialegEditarConducte`. Aquest diàleg accedeix a la funció `CablesDeConducte` del `GestorModelDB` per mostrar els cables relacionats amb aquest conducte si existeixen.
3. S'introdueixen o modifiquen les dades associades al conducte sel·leccionat. Es poden escollir les opcions:
 - (a) Acceptar. De manera que es crida la funció `ModificarConducte` de `DialegEditarConducte` que crida a `EditarConducte` de `GestorModelDB` amb les dades corresponents. Després es tanca el diàleg.
 - (b) Cancel·lar. Tanca directament el diàleg.
 - (c) Sel·leccionar un cable de la llista i pitjar el botó per editar-lo. Es crida la funció `EditarCable` de `GestorGIS` amb el cable sel·leccionat com paràmetre.
 - (d) Sel·leccionar un cable de la llista i pitjar el botó per eliminar-lo. Es crida la funció `EliminarCable` de `GestorModelDB` amb el cable sel·leccionat com paràmetre.
 - (e) Pitjar el botó per crear un nou cable. Es crida la funció `CrearCable` de `GestorModelDB`. Posteriorment es torna a llegir la funció `CablesDeConducte` del `GestorModelDB` per mostrar els cables relacionats amb aquest conducte. Es crida finalment a `EditarCable` de `GestorGIS` amb el cable creat com paràmetre.

Cas d'ús **Editar registre**

1. El *delineant* defineix el registre en el que desitja treballar.
 - (a) Traça un nou registre en la interfície GIS (espai de dibuix de *JUMP*). Un cop traçat aquest queda sel·leccionat per defecte.
 - (b) Altrament sel·lecciona el registre directament sobre la interfície GIS (espai de dibuix de *JUMP*).
2. Escull la comanda `EditarRegistre` de `GestorGIS`, accessible a través del menú o d'un botó de la barra d'eines incorporada a la interfície GIS. Això crea una instància de la classe `DialegEditarRegistre`. Aquest diàleg accedeix a la funció `CaixesDeRegistre` del `GestorModelDB` per mostrar les caixes relacionades amb aquest registre si existeixen i si existeix el Registre.
3. S'introdueixen o modifiquen les dades associades al registre sel·leccionat. Es poden escollir les opcions:
 - (a) Acceptar. De manera que es crida la funció `ModificarRegistre` de `DialegEditarRegistre` que crida a `EditarRegistre` de `GestorModelDB` amb les dades corresponents. Si el registre en concret no existeix encara es crea amb `CrearRegistre` de `GestorModelDB`. Després es tanca el diàleg.

- (b) Cancel·lar. Tanca directament el diàleg.
- (c) Sel·leccionar una caixa de la llista i pitjar el botó per editar-la. Es crida la funció `EditarCaixa` de `GestorGIS` amb la caixa sel·leccionada com paràmetre.
- (d) Sel·leccionar una caixa de la llista i pitjar el botó per eliminar-la. Es crida la funció `EliminarCaixa` de `GestorModelDB` amb la caixa sel·leccionada com paràmetre.
- (e) Pitjar el botó per crear una nova caixa. Es crida la funció `CrearCaixa` de `GestorModelDB`. Posteriorment es torna a llegir la funció `CaixesDeRegistre` del `GestorModelDB` per mostrar les caixes relacionades amb aquest registre. Es crida finalment a `EditarCaixa` de `GestorGIS` amb la caixa creada com paràmetre.

Cas d'ús **Editar cable**

1. El *delineant* escull el cable amb el que desitja treballar a partir d'una canalització per la que discorre el cable. Se sel·lecciona la canalització, el conducte i el cable corresponent amb els processos vistos en els casos d'ús **Editar canalització** i **Editar conducte**. En el `DialogEditarConducte` s'escull l'edició del cable sel·leccionat o la creació d'un nou cable.
2. Es crea una instància de la classe `DialogEditarCable`.
3. S'introdueixen o modifiquen les dades associades al cable sel·leccionat. Es poden escollir les opcions:
 - (a) Acceptar. De manera que es crida la funció `ModificarCable` de `DialogEditarCable` que crida a `EditarCable` de `GestorModelDB` amb les dades corresponents. Després es tanca el diàleg.
 - (b) Cancel·lar. Tanca directament el diàleg.

Cas d'ús **Editar caixa**

1. El *delineant* sel·lecciona el registre del que vol editar les caixes. El sel·lecciona directament sobre la interfície GIS, l'espai de dibuix de *JUMP*. Segueix el cas d'ús **Editar registre** fins arribar al punt en que es pitja el botó per editar la caixa sel·leccionada de la llista o la creació d'una nova caixa.
2. Es crea una instància de la classe `DialogEditarCaixa`. Aquest diàleg accedeix a la funció `CablesDeCaixa` del `GestorModelDB` per mostrar els cables relacionats amb aquesta caixa si existeixen.
3. S'introdueixen o modifiquen les dades associades a la caixa sel·leccionada. Es poden escollir les opcions:
 - (a) Acceptar. De manera que es crida la funció `ModificarCaixa` de `DialogEditarCaixa` que crida a `EditarCaixa` de `GestorModelDB` amb les dades corresponents. Després es tanca el diàleg.
 - (b) Cancel·lar. Tanca directament el diàleg.

- (c) Sel·leccionar un cable de la llista i pitjar el botó per afegir-lo. Es crida la funció `Afegir-Cable` de `GestorModelDB` amb el cable sel·leccionat com paràmetre. Posteriorment es torna a llegir la funció `CablesDeCaixa` del `GestorModelDB` per mostrar els cables relacionats amb aquesta caixa.
- (d) Sel·leccionar un cable de la llista i pitjar el botó per treure'l. Es crida la funció `Treure-Cable` de `GestorModelDB` amb el cable sel·leccionat com paràmetre. Posteriorment es torna a llegir la funció `CablesDeCaixa` del `GestorModelDB` per mostrar els cables relacionats amb aquesta caixa.

Cas d'ús **Editar empiulament**

1. El *gestorFO* sel·lecciona el registre del que vol editar els empiulaments de les caixes que conté. El sel·lecciona directament sobre la interfície GIS, l'espai de dibuix de *JUMP*.
2. Escull la comanda `EditarEmpiulament` de `GestorGIS`, accessible a través del menú o d'un botó de la barra d'eines incorporada a la interfície GIS. Això crea una instància de la classe `DialogEditarEmpiulament`.
3. S'introdueixen les dades associades a l'empiulament. Es poden escollir les opcions:
 - (a) **Empiular**. De manera que es crida la funció `CrearEmpiulament` de `DialogEditarEmpiulament` que crida a `CrearEmpiulament` de `GestorModelDB` amb les dades corresponents. Després es tanca el diàleg.
 - (b) **Trencar**. De manera que es crida la funció `TrencarEmpiulament` de `DialogEditarEmpiulament` que crida a `TrencarEmpiulament` de `GestorModelDB` amb les dades corresponents. Després es tanca el diàleg.
 - (c) **Tancar**. Tanca directament el diàleg.

Cas d'ús **Plotejar ruta**

1. El *supervisor* sel·lecciona el registre del que coneix passa la fibra òptica que porta el servei que desitja traçar sobre el mapa. Aquesta fibra passa per un cable d'una caixa. El sel·lecciona directament sobre la interfície GIS, l'espai de dibuix de *JUMP*.
2. Escull la comanda `PlotejarRuta` de `GestorGIS`, accessible a través del menú o d'un botó de la barra d'eines incorporada a la interfície GIS. Això crea una instància de la classe `DialogPlotejarRuta`.
3. S'introdueixen les dades associades a la fibra. Es poden escollir les opcions:
 - (a) **Dibuixar**. De manera que es crida la funció `PlotejarRuta` de `DialogPlotejarRuta` que crida a `PlotejarRutaFibra` de `GestorModelDB` amb les dades corresponents. Aquesta està implementada en llenguatge **PL/pgSQL** per optimitzar el motor de la base de dades i retorna tots els trams de la xarxa que contenen les fibres connectades a la sel·leccionada, de manera que poden sel·leccionar-se en l'espai de dibuix de *JUMP*. Després es tanca el diàleg.
 - (b) **Cancel·lar**. Tanca directament el diàleg.

4.3 Diagrama Estàtic de Disseny

Es pren com punt de partida el diagrama estàtic d'anàlisi (Figura 3.14) i s'afegeixen els mètodes descrits en el disseny dels casos d'ús. Igualment les classes s'han agrupat per subsistemes. En una primera revisió després de la descripció dels casos d'ús queda el diagrama de la Figura 4.3.

4.3.1 Disseny de la persistència

Pel disseny de la persistència es consideren els diagrames de classes de les Figures 3.4 i 3.6 i la utilització ja declarada del sistema gestor de bases de dades relacionals **PostgreSQL**.

Per fer el disseny de la base de dades primer cal eliminar l'herència, després s'elabora el diagrama d'Entitat — Relació i finalment s'especifica la base de dades.

Supressió de l'herència

No hi ha jerarquies d'herència en l'esquema de dades recollit en l'anàlisi.

Existeix si més no la implementació de les interfícies geomètriques de Tram i Punt, que impliquen afegir atributs geomètrics a les taules **Canalitzacio** i **Registre**. Aquesta adhesió es fa amb la instrucció `AddGeometryColumn()` de l'extensió **PostGIS** de **PostgreSQL** de la següent manera en el cas de **Canalitzacio**:

```
BEGIN WORK;

-- Table Canalitzacio is the way to connect the network model to the 'real' world
-- in the GIS representation. The table is defined with the non geometric
-- attributes and then the geometry is added with a PostGIS function.
-- SRID (Spatial Reference ID) is '0' for compatibility with Jump 1.1.1

SELECT AddGeometryColumn('public','canalitzacio','centerline','0','LINESTRING','2');

COMMIT;
```

Diagrama d'Entitat — Relació

El diagrama d'Entitat — Relació descriu el disseny conceptual del model obtingut en l'anàlisi. Es tenen en compte l'estructura i la integritat de les dades. Les entitats es representen per rectangles i les relacions entre aquestes per rombos.

En el cas de la interrelació entre **Canalitzacio** i **Registre** aquesta no és inclosa en l'esquema de dades sinó que, al tractar-se d'una relació topològica, serà tractada directament pel propi programari GIS. Aquesta relació fa referència a la coherència topològica de la malla o graf creat per la pròpia xarxa de canalitzacions i registres.

El diagrama es representa a la Figura 4.4.

La transformació del model Entitat — Relació cap al model relacional, és a dir el disseny lògic de les dades, té presents els següents principis [4]:

- Es transformen les entitats en taules o relacions del RDBMS.

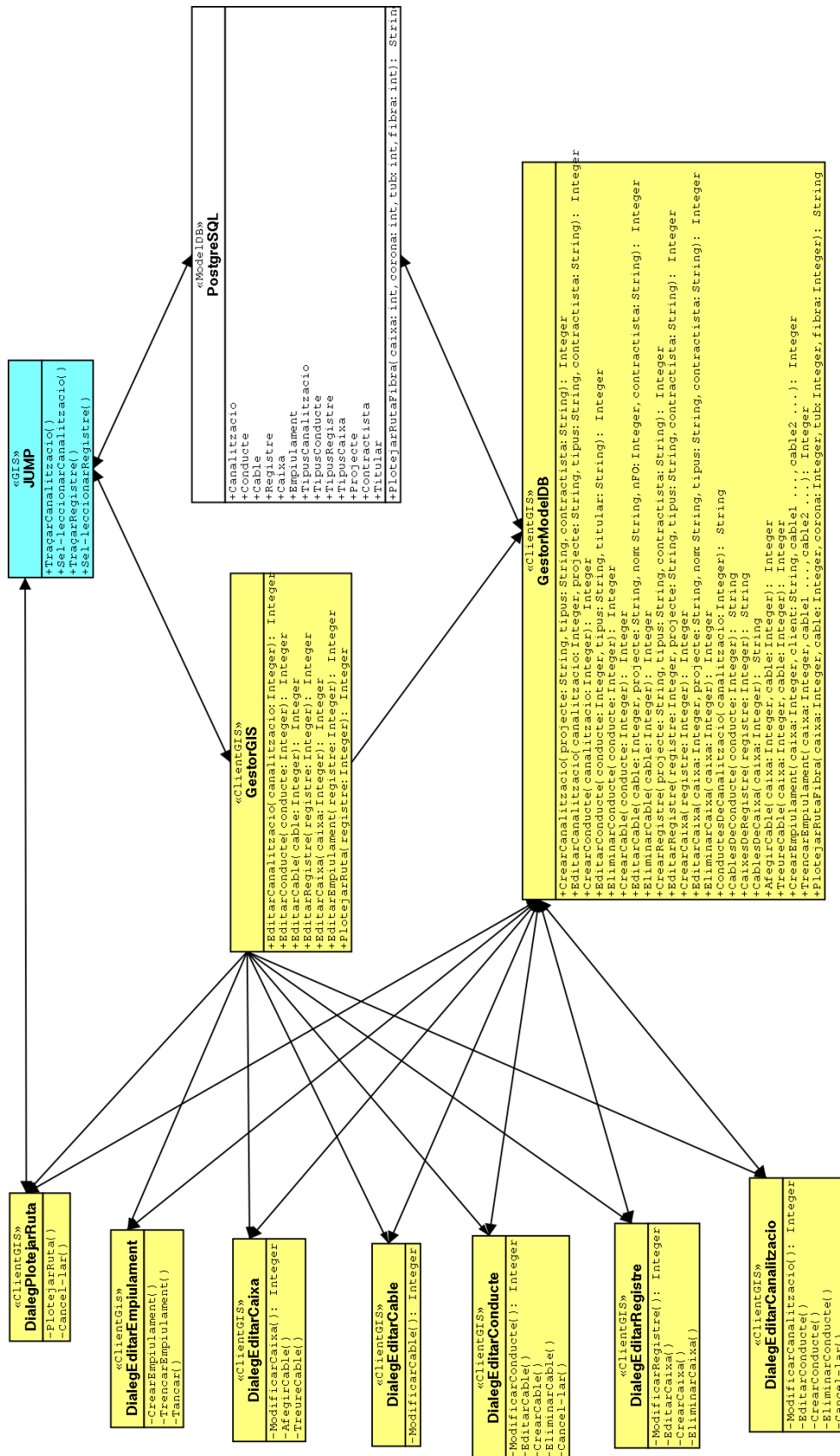


Figura 4.3: Diagrama estàtic de disseny en primera revisió

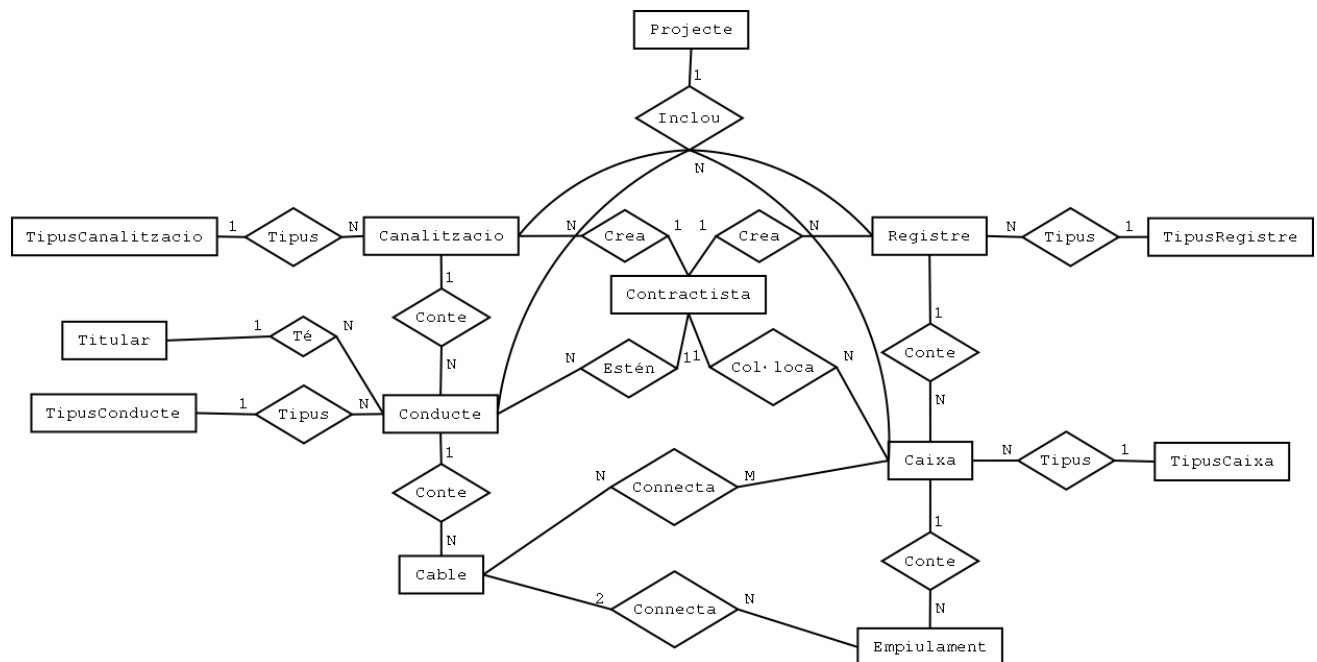
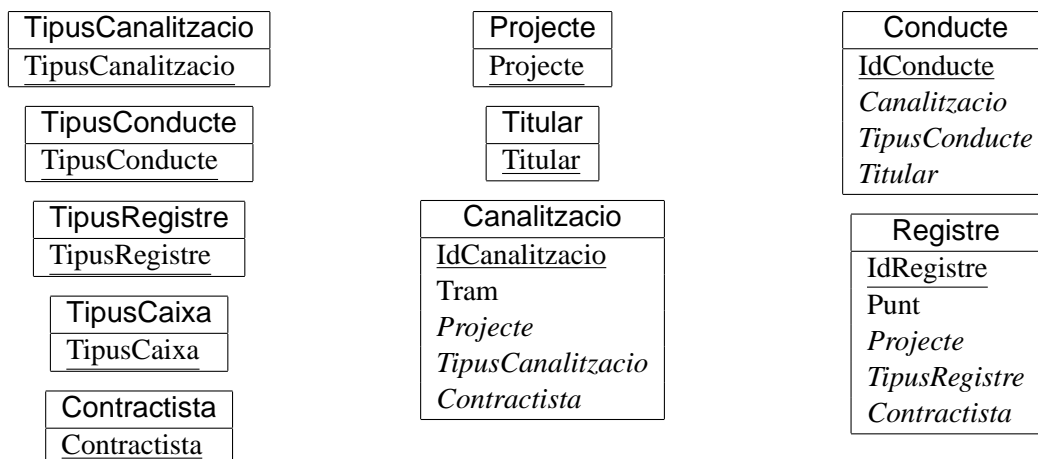
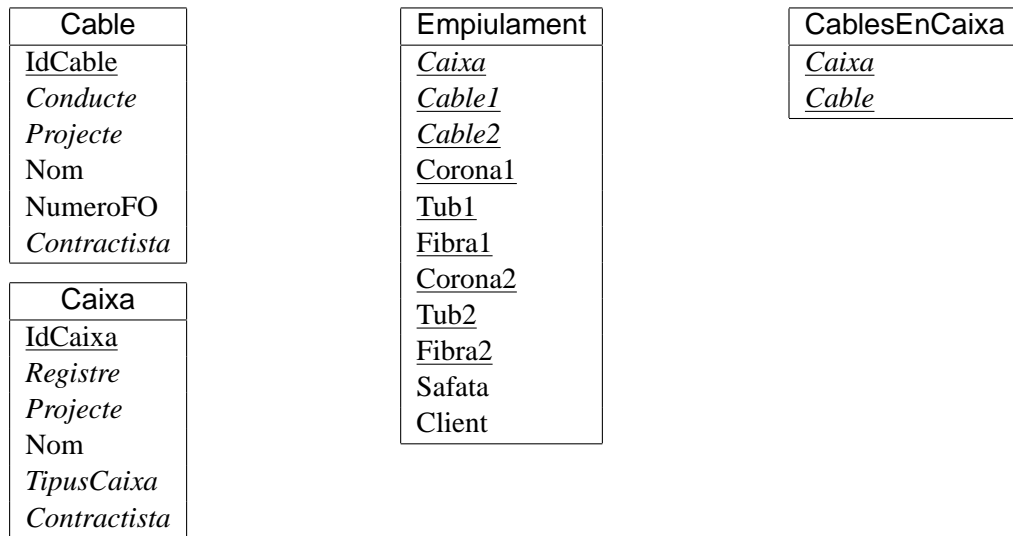


Figura 4.4: Model d'Entitat — Relació

- En les interrelacions de tipus $1 : N$ l'entitat de la banda N adquireix una clau forana que referència l'altra entitat.
- En les interrelacions de tipus $N : M$ aquesta es transforma en una taula o relació del RDBMS amb les claus foranes que referencien a cada una de les dues entitats que relaciona.

D'aquesta forma les diferents entitats queden amb els següents atributs, on es representen de manera que resaltin les claus primàries i les claus *foranes*. En general la transformació és senzilla, excepte en el cas de la interrelació entre **Cable** i **Empiulament** on es tracta d'una relació $2 : N$ que provoca la duplictat de codis forans.





Especificació

Finalment es declaren en forma de pseudocodi, pràcticament codi SQL, les especificacions de les diferents taules o relacions de la base de dades. Cal tenir present que **PostgreSQL** permet la generació directa de seqüències que es poden fer servir per autogenerar claus, en aquest cas es crea una per utilitzar un valor únic de clau sigui quin sigui l'element, és a dir que si una *Canalitzacio* té un valor de la seva clau primària *X*, cap altre element de l'esquema de dades té aquest valor en una clau primària.

```
-- Create sequence. Starts so far to work with sample data with no interference
-- with the primary keys of data inserted by hand.

CREATE SEQUENCE Ids_seq MINVALUE 1000000;

-- Create tables:

-- Create domains for fields in main tables.

BEGIN WORK;

CREATE TABLE TipusCanalitzacio (
    TipusCanalitzacio varchar (10) PRIMARY KEY
);

CREATE TABLE TipusConducte (
    TipusConducte varchar (10) PRIMARY KEY
);

CREATE TABLE TipusRegistre (
    TipusRegistre varchar (10) PRIMARY KEY
);

CREATE TABLE TipusCaixa (
    TipusCaixa varchar (64) PRIMARY KEY
);

CREATE TABLE Titular (
    Titular varchar (64) PRIMARY KEY
);
```

```
CREATE TABLE Contractista (  
    Contractista varchar (64) PRIMARY KEY  
);  
  
CREATE TABLE Projecte (  
    Projecte varchar (10) PRIMARY KEY  
);  
  
COMMIT;  
  
-- Create main tables.  
  
BEGIN WORK;  
  
CREATE TABLE Canalitzacio (  
    IdCanalitzacio int DEFAULT nextval('Ids_seq') PRIMARY KEY,  
    Projecte varchar NOT NULL,  
    TipusCanalitzacio varchar NOT NULL,  
    Contractista varchar NOT NULL,  
  
    FOREIGN KEY (Projecte) REFERENCES Projecte ON UPDATE CASCADE,  
    FOREIGN KEY (TipusCanalitzacio) REFERENCES TipusCanalitzacio ON UPDATE CASCADE,  
    FOREIGN KEY (Contractista) REFERENCES Contractista ON UPDATE CASCADE  
);  
  
CREATE TABLE Conducte (  
    IdConducte int DEFAULT nextval('Ids_seq') PRIMARY KEY,  
    Canalitzacio int NOT NULL,  
    TipusConducte varchar NOT NULL,  
    Titular varchar NOT NULL,  
  
    FOREIGN KEY (Canalitzacio) REFERENCES Canalitzacio ON DELETE CASCADE,  
    FOREIGN KEY (Titular) REFERENCES Titular ON DELETE CASCADE,  
    FOREIGN KEY (TipusConducte) REFERENCES TipusConducte ON UPDATE CASCADE  
);  
  
CREATE TABLE Cable (  
    IdCable int DEFAULT nextval('Ids_seq') PRIMARY KEY,  
    Conducte int NOT NULL,  
    Projecte varchar NOT NULL,  
    NumeroFO int NOT NULL,  
    Contractista varchar NOT NULL,  
  
    FOREIGN KEY (Conducte) REFERENCES Conducte ON DELETE CASCADE,  
    FOREIGN KEY (Projecte) REFERENCES Projecte ON UPDATE CASCADE,  
    FOREIGN KEY (Contractista) REFERENCES Contractista ON UPDATE CASCADE,  
    CHECK (NumeroFO >= 0)  
);  
  
CREATE TABLE Registre (  
    IdRegistre int DEFAULT nextval('Ids_seq') PRIMARY KEY,  
    Projecte varchar NOT NULL,  
    TipusRegistre varchar NOT NULL,  
    Contractista varchar NOT NULL,  
  
    FOREIGN KEY (Projecte) REFERENCES Projecte ON UPDATE CASCADE,  
    FOREIGN KEY (TipusRegistre) REFERENCES TipusRegistre ON UPDATE CASCADE,  
    FOREIGN KEY (Contractista) REFERENCES Contractista ON UPDATE CASCADE  
);  
  
CREATE TABLE Caixa (  
    IdCaixa int DEFAULT nextval('Ids_seq') PRIMARY KEY,  
    Registre int NOT NULL,  
    Projecte varchar NOT NULL,  
    TipusCaixa varchar NOT NULL,  
    Nom varchar NOT NULL,  
    Contractista varchar NOT NULL,
```

```

    FOREIGN KEY (Registre) REFERENCES Registre ON DELETE CASCADE,
    FOREIGN KEY (Projecte) REFERENCES Projecte ON UPDATE CASCADE,
    FOREIGN KEY (TipusCaixa) REFERENCES TipusCaixa ON UPDATE CASCADE,
    FOREIGN KEY (Contractista) REFERENCES Contractista ON UPDATE CASCADE
);

CREATE TABLE Empiulament (
    Caixa int NOT NULL,
    Cable1 int NOT NULL,
    Cable2 int NOT NULL,
    Coronal int NOT NULL,
    Corona2 int NOT NULL,
    Tub1 int NOT NULL,
    Tub2 int NOT NULL,
    Fibras1 int NOT NULL,
    Fibras2 int NOT NULL,
    Safata int NOT NULL,
    Client varchar NOT NULL,

    PRIMARY KEY (Caixa, Cable1, Cable2, Coronal, Corona2, Tub1, Tub2, Fibras1, Fibras2),
    FOREIGN KEY (Caixa) REFERENCES Caixa ON DELETE CASCADE,
    FOREIGN KEY (Cable1) REFERENCES Cable ON DELETE CASCADE,
    FOREIGN KEY (Cable2) REFERENCES Cable ON DELETE CASCADE
);

CREATE TABLE CablesEnCaixa (
    Cable int NOT NULL,
    Caixa int NOT NULL,

    FOREIGN KEY (Cable) REFERENCES Cable ON DELETE CASCADE,
    FOREIGN KEY (Caixa) REFERENCES Caixa ON DELETE CASCADE
);

COMMIT;
```

Els atributs geomètrics s'afegeixen posteriorment amb la comanda específica, tal com ja s'ha mostrat abans en parlar de la utilització de les interfícies geomètriques que proporciona **PostGIS**.

4.3.2 Subsistemes

El disseny de les diferents interaccions entre els subsistemes es recull ara de forma cronològica de manera que es segueix el procediment per crear des de zero tota la connectivitat. Primer es crea la base de dades incloent-hi la funcionalitat geomètrica i posteriorment es llença el sistema GIS amb la connectivitat a la base de dades geomètrica i el mòdul del *ClientGIS* en marxa.

ModelDB

Es considera com punt de partida una compilació recent de **PostgreSQL** del qual es té el codi font en un directori adient. En aquest codi font s'ha afegit, tal com expressa en les instruccions d'instal·lació, el codi de l'extensió **PostGIS** (normalment en el directori `/share/contrib` de **PostgreSQL**).

Cal definir una política d'accés al servidor de dades útil per les circumstàncies en les que es trobi. Es llença el servidor de dades, per defecte farà servir el port 5432, normalment amb la comanda:

```
/usr/local/pgsql/bin/postmaster -i -D /usr/local/pgsql/data/
```

Es crea la base de dades en el servidor i es carrega la capacitat geomètrica.

```
DBNAME=NomDB
PGSUPERUSER=postgres
PGDIR=/usr/local/pgsql/bin
POSTGISDIR=/usr/local/pgsql/share/contrib
DATADIR=/home/santiago/Documents/dev/ModelDB

$PGDIR/createdb $DBNAME
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -q -f ' $POSTGISDIR/postgis.sql
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -q -f ' $POSTGISDIR/spatial_ref_sys.sql
```

Posteriorment es carrega l'esquema de taules i seqüències. El codi de l'esquema és molt semblant al presentat abans per la especificació de la persistència. També es carreguen les funcions de cerca definides directament en codi **PL/pgSQL** per optimitzar-les.

```
$PGDIR/psql -d $DBNAME -q -f $DATADIR/ModelDB.sql
$PGDIR/psql -d $DBNAME -q -f $DATADIR/CodiPL.sql
```

La base de dades està llesta per rebre dades del sistema GIS i interactuar-hi.

ClientGIS

El llançament de **JUMP** necessita de diverses classes que s'exposen en el classpath:

```
.../lib/xerces.jar
.../lib/Acme.jar
.../lib/JTS_Test.jar
.../lib/jdom.jar
.../lib/jts-1.4.0.jar
.../lib/Jama-1.0.1.jar
.../lib/pg74jdbc3.jar
```

Igualment necessita declarar que es farà servir l'extensió **PostGISDriver** per l'accés a dades geomètriques proporcionades per **PostGIS**. Aquesta circumstància pot passar-se com paràmetre del propi GIS en la línia de comandes. També es carrega de la mateixa forma, tot i que es podria fer a través del classpath amb un mòdul .jar, el propi *ClientGIS*.

```
<workbench>
  <plug-in>
    net.refractions.postgis.PostGISPlugin
  </plug-in>
  <plug-in>
    uoc.tfc.ClientGIS
  </plug-in>
</workbench>
```

Un cop llençat **JUMP** es troba en disposició de connectar amb la base de dades i interoperar amb normalitat.

Diagrama Estàtic de Disseny

Finalment el diagrama estàtic queda segons la Figura 4.5. S'inclouen les funcions definides prèviament i l'esquema de dades del *ModelDB*.

4.4 Interfície

La interfície es separa en els diferents subsistemes que admeten interacció amb l'usuari:

- La interfície de *ClientGIS* es manté molt similar a l'analitzada en el capítol anterior. S'ha afegit els botons per editar/eliminar/crear els diferents elements de les llistes. En el cas de *Dialeg-EditarCaixa* i *DialegEditarEmpiulament* s'ha adaptat millor al disseny dels casos d'ús que els utilitzen. Apareix també un nou element, una botonera que permet l'accés a les accions o comandes activades des dels casos d'ús **Editar canalització**, **Editar registre**, **Editar empiulament** i **Plotejar ruta**.
- La interfície de **JUMP** es mostra en diverses captures de pantalla. D'una banda es mostra l'entorn de dibuix i d'altra el diàleg de connexió a través del **PostGISDriver** contra la base de dades.
- La interfície de la base de dades mostra el tractament des de consola del programari **PostgreSQL**.

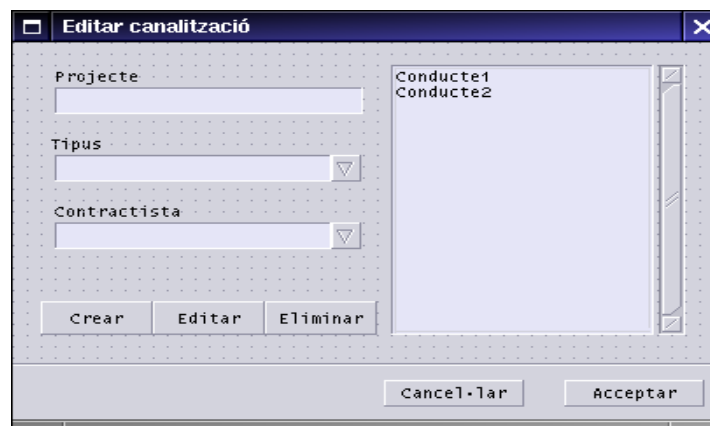


Figura 4.6: Interfície de la classe DialogEditarCanalitzacio

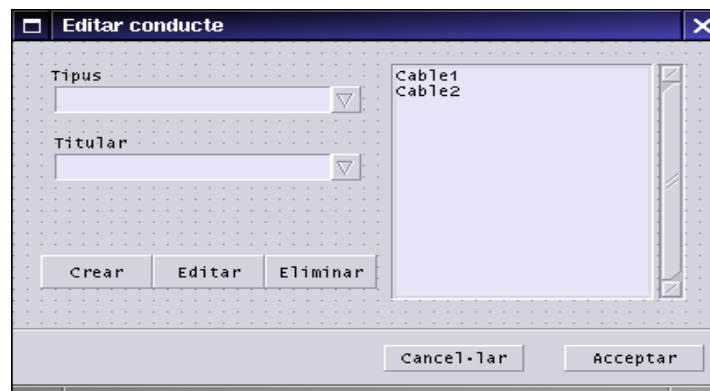


Figura 4.7: Interfície de la classe DialogEditarConducte

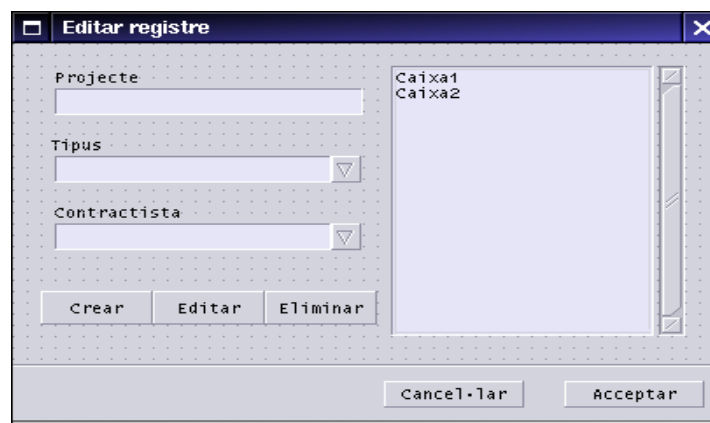


Figura 4.8: Interfície de la classe DialogEditarRegistre

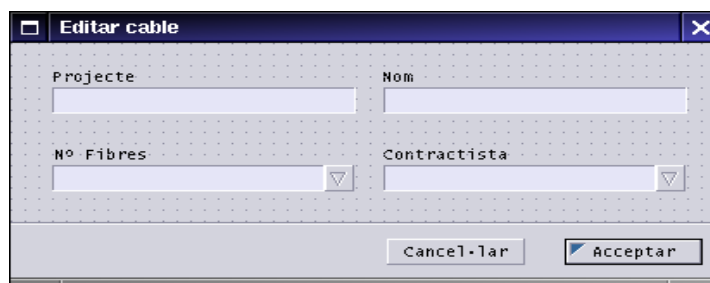


Figura 4.9: Interfície de la classe DialogEditarCable

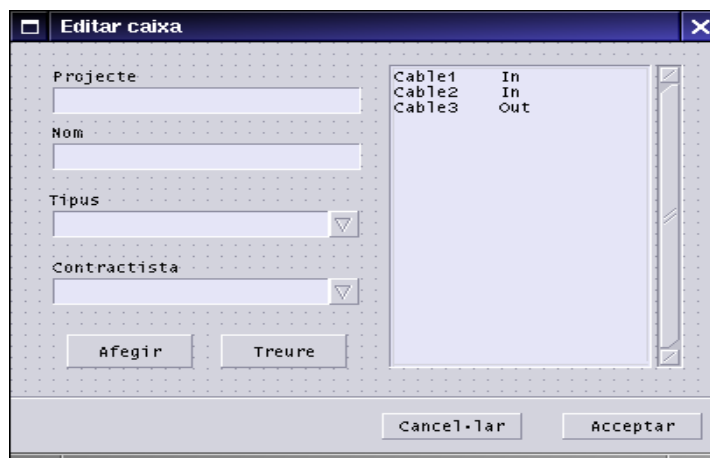


Figura 4.10: Interfície de la classe DialogEditarCaixa

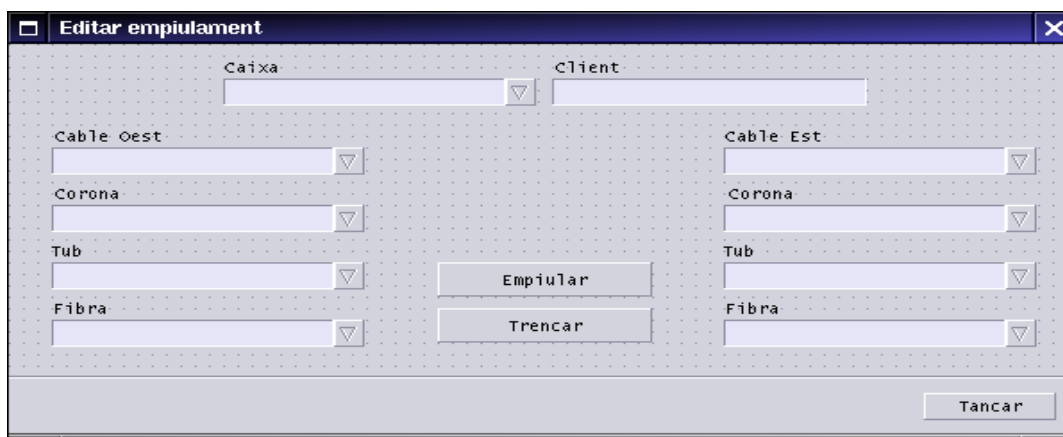


Figura 4.11: Interfície de la classe DialogEditarEmpiulament



Figura 4.12: Interfície de la classe DialegPlotejarRuta

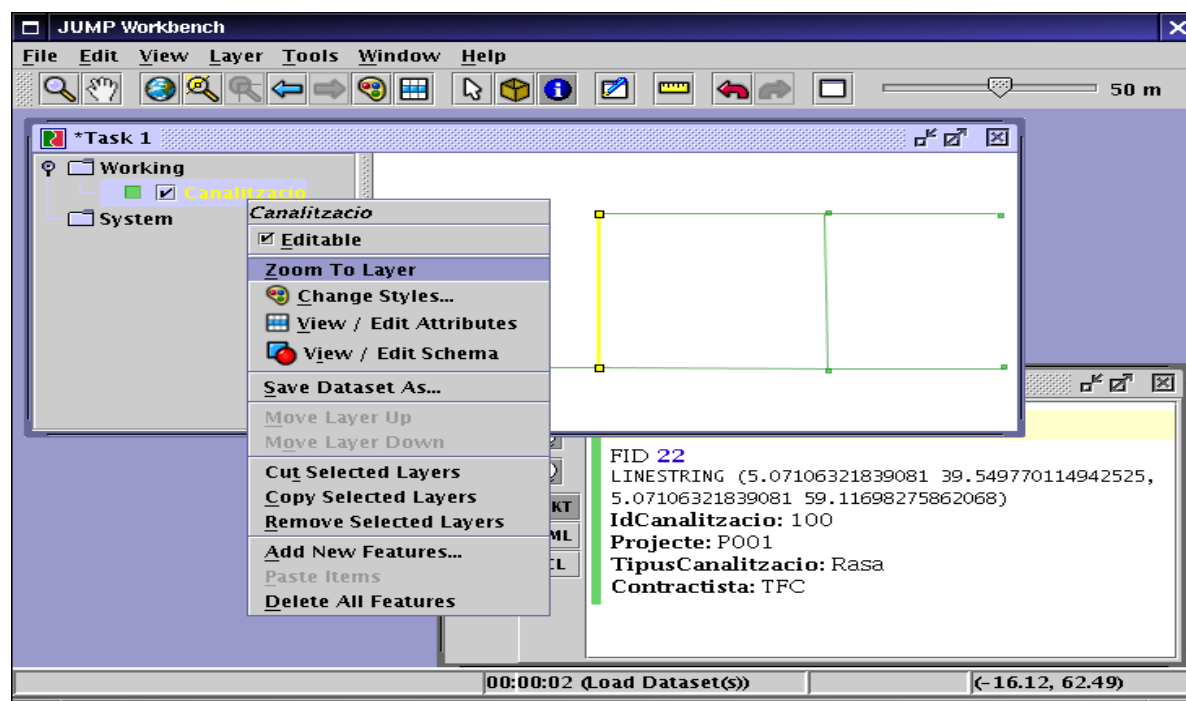
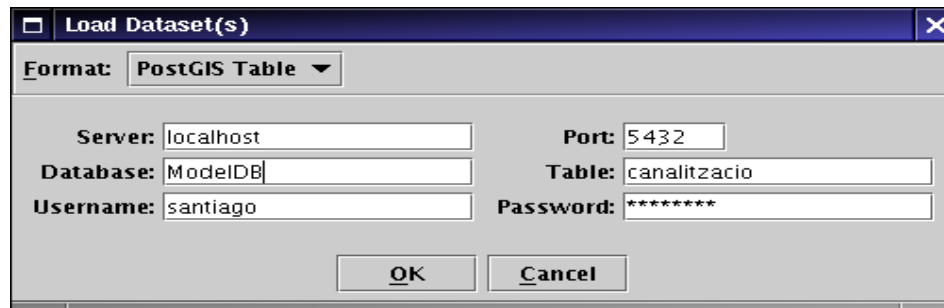
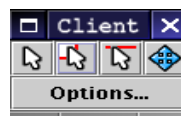
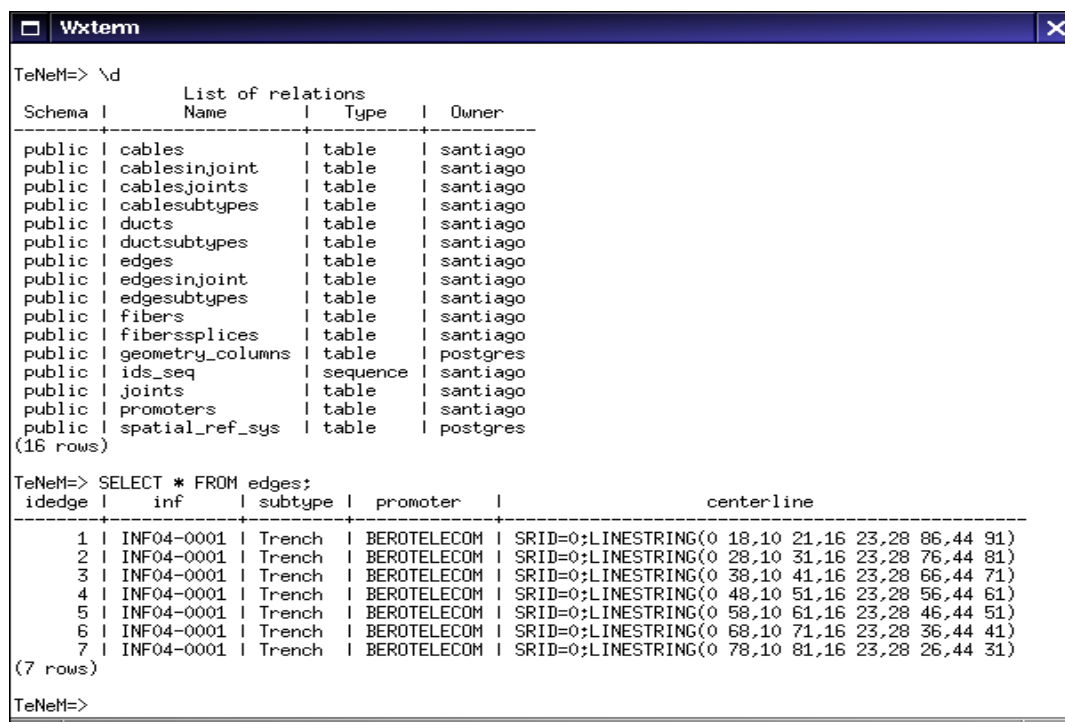


Figura 4.13: Interfície de dibuix de JUMP

Figura 4.14: Interfície de **PostGISDriver**Figura 4.15: Interfície de comandes *ClientGIS*Figura 4.16: Interfície de **PostgreSQL**

Capítol 5

Valoració econòmica

La utilització de programari lliure [8] és una opció que actualment genera una forta polèmica per la seva valoració en termes econòmics. Existeixen molts estudis que afavoreixen el programari lliure davant del propietari i molts d'altres que demostren que la tendència és contrària.

Utilitzar programari lliure provoca un cost de llicència nul, doncs el programari és adquirible amb el codi font de forma gratuïta. En la implantació de programari cal tenir present però d'altres costos com el d'exploració i manteniment, que en aquest cas poden arribar a ser més elevats que en el programari propietari.

Davant d'això i considerant com premissa un cost similar d'ambdues opcions cal tenir present el factor llibertat que proporciona el programari lliure; d'una banda el desenvolupament de noves funcionalitats es pot fer a instàncies de l'usuari quan aquest ho cregui convenient i en el cas de necessitar canviar d'aplicatiu per fallida de l'utilitzat en un moment determinat (abandonament del desenvolupament, aparició d'un nou aplicatiu més evolucionat o amb funcionalitats desitjades) aquesta translació és relativament senzilla pel perfecte coneixement dels formats interns dels arxius d'informació.

En definitiva el programari lliure passa a ser rendible econòmicament si els responsables dels sistemes d'informació tenen coneixements amplis en la seva instal·lació, configuració i utilització. La resta del cost de desenvolupament del model i programari client són equivalents que amb la utilització de programari propietari. Això permet que l'inici del desenvolupament tingui un cost mínim, directament proporcional a la feina realitzada, sense haver d'invertir en llicències ni maquinari massa car.

Capítol 6

Conclusions

La tasca de definició de com ha de ser l'estructura de dades d'un model de xarxa de telecomunicacions s'ha assolit, i s'ha fet tenint en compte les necessitats funcionals immediates que tindran els usuaris principals d'aquesta aplicació, fins i tot amb el disseny de la interfície d'usuari des d'un aplicatiu GIS.

Aquest model i aquest client de l'aplicatiu GIS s'han dissenyat tenint en compte aspectes concrets que es requereixen en l'entorn d'una empresa de telecomunicacions, però aquests requeriments en realitat són molt més amplis i complexes. És probable que el disseny aconseguit en aquest treball sigui suficient per iniciar la construcció d'un aplicatiu que doni resposta a les necessitats plantejades, però també és probable que apareguin, en el desenvolupament d'aquest aplicatiu, noves idees que obliguin a ampliar i fins i tot a modificar aspectes fonamentals del model de dades obtingut.

El fet de publicar amb llicència oberta el codi que ha crescut en paral·lel a aquest treball i del que ha agafat algunes idees, permet mostrar a qui estigui interessat els aspectes de detall d'aquest desenvolupament, alhora que es manté certa garantia sobre els drets d'autor.

En el futur i sempre que el temps material ho permeti, se seguirà desenvolupant el prototipus **TeNeM** [9] de manera que s'aconsegueixi la funcionalitat desitjada en aquest treball i noves facilitats com:

- Gestió d'espais lliures en conductes.
- Anàlisi de punts crítics de xarxa.
- Assistent a traçador de rutes de fibra.
- Localitzador d'avaries en xarxa de fibra.

Finalment vull agrair a la meva dona Paloma la paciència que ha tingut amb mi al llarg de tota la carrera d'Enginyeria Tècnica en Informàtica de Gestió a la Universitat Oberta de Catalunya que culmina amb la presentació d'aquest treball. El seu suport i ajuda en els moments de càrrega i desorientació són el millor revifador.

Capítol 7

Glossari

7.1 Introducció

Descripció dels elements més comuns d'una xarxa de telecomunicacions de fibra òptica i que són modelitzats al programari. L'objecte d'aquesta recopilació és la definició de les diferents parts de la xarxa que s'han modelitzat a fi i terme de fer més entenedor el model i d'unificar criteris.

També es defineixen diferents conceptes que apareixen en el treball i que habitualment són desconeguts o que necessiten una definició clara i concreta dins de l'àmbit del treball.

7.2 Elements lineals

Els elements lineals (trams) de la xarxa són:

7.2.1 Canalització

Les canalitzacions són trams de xarxa que corresponen a una rasa més o menys recta que uneix dos registres. L'execució física de les canalitzacions consta de l'excavació de la rasa, la col·locació de conductes pel pas de cables, la protecció dels conductes generalment amb formigó i el posterior reomplert de la rasa i reposició del paviment.



Figura 7.1: Esquema transversal de canalització de telecomunicacions [5]

La linealitat de la rasa és important per la posterior estesa de cables; si s'executa amb massa curvatura o amb ziga zagues, en el moment de l'estesa la fricció dels cables amb les parets dels tubs

serà massa forta i pot impedir aquesta o fins i tot trencar el cable.

7.2.2 Conductes

Les canalitzacions estan formades normalment per un prisma de conductes preparats pel pas de cables. Cada conducte o tubular, normalment de 125mm o 40mm de diàmetre, permet el pas de diversos cables pel seu interior, de manera que és possible la manipulació i estesa dels cables un cop executada la canalització.

Els tubs estan disposats de manera ordenada dins del prisma, normalment de forma que s'estalviï recursos d'excavació o amb la combinació que permetin les característiques del terreny. El nombre de tubs habitual és de 2, 4, 6, 9, 12 o superiors.

7.2.3 Cables

Els cables de fibra òptica habituals són relativament manipulables i lleugers. Els seus diàmetres varien d'un a tres centímetres. Normalment estan formats per una coberta plàstica amb més o menys capes i uns petits conductes al seu interior que contenen les fibres. Aquests tubs poden estar disposats en diverses capes (corones) segons el disseny del cable. Hi ha cables especials per usos determinats que afegeixen cobertes d'acer, d'aramida, etc. També hi ha cables molt especials com els submarins per telecomunicacions transoceàniques.



Figura 7.2: Cable de fibra òptica [12]

7.2.4 Fibres

Les fibres òptiques són fibres del tamany d'un cabell humà que arriben a tenir quilòmetres de longitud. Estan diferenciades en un nucli i una coberta. Pel nucli és per on circula la llum reflexant-se per les parets gràcies a les propietats del material, normalment una vitroresina. Aquesta llum transporta el senyal de telecomunicacions.

La senyal pateix una pèrdua amb la distància recorreguda per la fibra, el que provoca la necessitat de repetidors de senyal cada certa distància (desenes de quilòmetres) segons les propietats del material i la font de llum. Igualment es pateix pèrdua de senyal si el radi de gir al que està sotmesa la fibra és excessiu. En els punts on hi ha canvis d'estat també resulten importants pèrdues de senyal, com per exemple les connexions per fusió i amb més intensitat les connexions per elements que permeten la manipulació, normalment anomenats 'connectors'.

Les fibres doncs es connecten entre elles al encarar-les una davant de l'altra. En aquesta posició es



Figura 7.3: Fibra òptica [13]

pot procedir a la fusió de les fibres, creant un ‘empiulament’ permanent i que només pot desconnectar-se per rotura, o es col·loquen uns petits acoblaments terminals que permeten la manipulació, connectant i desconnectant, anomenats “connectors”.

7.3 Elements puntuals

Els elements puntuals (vèrtexs) de la xarxa són:

7.3.1 Registre

El registres són punts de la xarxa on és possible la inspecció i manipulació dels cables i de les caixes d’empiulaments que s’ubiquen al seu interior. Físicament corresponen a les cambres, pericons, arquetes i altres construccions que articulen les canalitzacions i permeten en algunes ocasions fins i tot l’accés al seu interior del personal degut a les dimensions que tenen.



Figura 7.4: Registre prefabricat [11]

Degut a les característiques de la xarxa la seva existència és obligatòria, doncs permeten fer canvis de direcció a les canalitzacions (aquestes han de tenir una traçada el més lineal possible) i també són els punts per on es fan les esteses de cables i on queden emplaçades les caixes d’empiulaments que connecten els cables.

7.3.2 Caixa d’empiulaments

Les caixes d’empiulaments són dispositius que intercepten la xarxa de cables per tal de permetre realitzar empiulaments de fibres òptiques en el seu interior. D’aquesta manera en una caixa poden entrar dos o més cables i es poden desviar rutes entre les fibres d’aquests cables. Físicament són normalment de materials plàstics i resistent i les seves mides depenen de la capacitat d’empiulaments

que poden arribar a mantenir al seu interior. En general tenen un tamany que permet ubicar-les dins dels registres.



Figura 7.5: Caixa d'empulaments [14]

L'estructura interna de les caixes d'empulament és força complicada i depés dels models i fabricants. En general totes tenen unes 'safates' on van a parar els tubs nus que formen els cables de fibra òptica i on es despenen aquestes fibres. En aquestes safates hi ha els mecanismes de suport dels diferents empulaments de fusió que es realitzen.

Hi ha també caixes metàl·liques o caixes que contenen connectors al seu interior.

7.3.3 Empulaments

Els empulaments són les connexions entre fibres que es realitzen per fusió. Amb un aparell especial s'encaren les dues puntes de fibra a empular i es provoca un arc elèctric que fusiona i dona continuïtat a la fibra. Si la fusió queda ben feta les pèrdues de senyal que pateix la llum en l'empulament són pràcticament inapreciables, és a dir que la continuïtat queda perfecta.



Figura 7.6: Empulament de fibra òptica [15]

Cal tenir present que els empulaments són un punt molt delicat de la ruta de fibres, de manera que aquests es protegeixen per unes peces especials de plàstic termoretràctil i es col·loquen ben subjectades dins de les caixes d'empulaments.

Bibliografia

- [1] Benet Campderrich Falgueras, *Enginyeria del Programari I*, Fundació per la Universitat Oberta de Catalunya, Dipòsit legal B-37.646-00, ISBN 84-8429-121-9.
- [2] Ivar Jacobson, Grady Booch and James Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 1999, ISBN 0-201-57169-2.
- [3] Desmond Francis D'Souza and Alan Cameron Wills, *Objects, Components and Frameworks with UML. The CatalysisSM Approach*, Addison Wesley, 1999, ISBN 0-201-31012-0
- [4] Jaume Sistac Planas, *Bases de Dades I*, Fundació per la Universitat Oberta de Catalunya, Dipòsit legal B-27.451-2002, ISBN 84-8429-586-9.
- [5] REAL DECRETO 401/2003, de 4 de abril, por el que se aprueba el *Reglamento regulador de las infraestructuras comunes de telecomunicaciones para el acceso a los servicios de telecomunicaciones en el interior de los edificios y de la actividad de instalación de equipos y sistemas de telecomunicaciones*. B.O.E., nº 115, de 14 de mayo de 2003, págs. 18459 a 18502.
- [6] Pere Soler Matutes, *El Contrato para la Elaboración de Programas de Ordenador, Contrato de desarrollo de software*, Aranzadi, Dipòsit legal 3282/2003, ISBN 84-9767-321-2
- [7] Jesús González Barahona, Joaquín Seoane Pascual y Gregorio Robles, *Introducción al Software Libre*, Fundació per la Universitat Oberta de Catalunya, Dipòsit legal B-38.682-2003, ISBN 84-9788-028-5.
- [8] GNU Operating System, Free Software Foundation, *Free as in Freedom* [on line] <<http://www.gnu.org>> [1 juny 2004]
- [9] Santiago Payà i Miralta, *TeNeM Telecommunications Network Model* [on line] <<http://www.sourceforge.net/tenem>> [5 abril 2004]
- [10] L^AT_EX, *A document preparation system* [on line] <<http://www.latex-project.org>> [1 juny 2004]
- [11] FORTE Fabricant, *Figures de registre prefabricat* [on line] <<http://www.forte.es>> [5 abril 2004]
- [12] ALCATEL, *Figures de cable* [on line] <<http://www.alcatel.com>> [5 abril 2004]
- [13] HOWSTUFFWORKS, *Figures de fibra* [on line] <<http://www.howstuffworks.com>> [5 abril 2004]

-
- [14] TELECOMOSP, *Figures de caixes d'empilaments* [on line] <<http://www.telecomosp.com>> [5 abril 2004]
- [15] LANS HACK, *Figures d'empilaments de fibra òptica* [on line] <<http://www.lanshack.com>> [5 abril 2004]
- [16] ORACLE, *Oracle Corporation* [on line] <<http://www.oracle.com>> [1 maig 2004]
- [17] ESRI, Environmental Systems Research Institute, *GIS and Mapping Software* [on line] <<http://www.esri.com>> [1 maig 2004]
- [18] Dia *Dia a drawing program* [on line] <<http://www.gnome.org/projects/dia/>> [5 abril 2004]
- [19] Glade, *GTK+ User Interface Builder* [on line] <<http://glade.gnome.org/>> [5 abril 2004]
- [20] PostgreSQL, *The most advanced Open Source database system in the world* [on line] <<http://www.postgresql.org>> [1 maig 2004]
- [21] JUMP, Unified Mapping Platform [on line] <<http://www.vividsolutions.com/jump>> [1 maig 2004]
- [22] PostGIS, *Geographics objects for PostgreSQL* [on line] <<http://www.postgis.org>> [1 maig 2004]
- [23] PostGIS JUMP Driver, *JUMP 1.1 plug-in will provide basic support for reading and writing from a PostGIS database* [on line] <<http://postgis.refractions.net>> [1 maig 2004]
- [24] Vivid Solutions, Inc., *Software development company that creates and implements innovative solutions for a variety of computing needs* [on line] <<http://www.vividsolutions.com>> [1 juny 2004]
- [25] Refrations Research, Inc., *Consulting and product development organization, dedicated to providing great technology and a great client experience* [on line] <<http://www.refrations.net/>> [1 juny 2004]

Part II

Annexes

Capítol 8

Pla de treball

S'adjunta el pla de treball que presenta les intencions davant del tutor del projecte abans de l'inici d'aquest. S'ha corregit algunes faltes ortogràfiques i l'esmena de l'OMG com autor del RUP.

8.1 Descripció del Treball Fi de Carrera

Títol: Definició de Requeriments, Anàlisi i Disseny del Model de Dades i el Funcionament associat per Gestionar una Xarxa de Telecomunicacions a través d'un Sistema d'Informació Geogràfica.

Es tracta de desenvolupar les fases lògiques per la creació d'un aplicatiu que doni resposta als requeriments dels usuaris que gestionen una xarxa de telecomunicacions. Aquestes fases són:

1. Recollida dels requeriments.
2. Anàlisi dels requeriments.
3. Disseny d'una solució.
4. Implementació en codi de la solució dissenyada.

L'última fase només es durà a terme si el temps i les fases anteriors permeten una codificació relativament senzilla.

8.1.1 Objectius Generals i Específics

El problema al que s'ha de donar solució és la gestió d'una xarxa de telecomunicacions per fibra òptica. Aquesta està composta normalment per trams de canalització en rasa amb conductes al seu interior que alhora contenen cables, aquests cables tenen un nombre determinat de fibres òptiques en la seva composició. Els diferents trams formen una xarxa que ens els vèrtexs normalment tenen pericons o cambres de registre. L'abstracció de la xarxa dóna un graf diferent segons si tractem la xarxa de canalitzacions, la de tubs, la de cables o la de fibres, però cada graf s'extrau en part del graf anterior.

Els principals reptes de la gestió a assolir són:

- Creació del model de dades que doni suport a la xarxa completa, amb tots els seus elements.
- Manteniment de la consistència del model.
- Càlculs analítics de la xarxa com per exemple el càlcul de rutes òptimes o el càlcul de punts crítics.
- Altres que sorgeixin de la recollida i anàlisi de requeriments.

Un primer esboç de solució podria estar orientat cap a una implementació de l'estil:

1. Crear un model de dades en un SGBD de manera que respongui els requeriments detectats.
2. Crear les rutines que donin la funcionalitat desitjada amb PL/SQL o a través de connexions al SGBD com ODBC/JDBC o directament amb llibreries C.
3. Connectar el model i la seva funcionalitat a un sistema GIS que mantingui l'estàndard de l'Open Gis Consortium (<http://www.opengis.org>) a través d'una interfície adequada o com un mòdul del propi aplicatiu GIS.

8.1.2 Planificació

Les diferents tasques a desenvolupar i el calendari associat es proposen com:

Tasca	Durada	Data inici	Data finalització
Planificació de tasques	7	03/03/04	10/03/04
Recollida de requeriments	13	10/03/04	23/03/04
Anàlisi dels requeriments	27	17/03/04	13/04/04
Disseny de la solució	33	14/04/04	17/05/04
Lliurament Final	31	18/05/04	18/06/04

Les principals dates es marquen en el calendari (de la web de l'aula):

Alhora que es porten a terme aquestes fites s'anirà desenvolupant al llarg del treball:

- La codificació sempre que sigui possible per fer proves o la pròpia implementació final.
- Comunicació continua amb el consultor.

8.1.3 Model i Mètode

El model a utilitzar serà l'orientat a objectes. Es tracta de desenvolupar una abstracció d'uns ens existents en la realitat i que entre d'altres atributs mantenen uns comportaments i unes relacions entre ells que fan més adequat seguir aquesta orientació que no pas l'estructurada, potser més adequada si ens centréssim en la funcionalitat i la gestió que no pas en la pròpia xarxa. Se seguirà l'estàndard Unified Modeling Language (UML) de l'Object Management Group (<http://www.omg.org>).

El mètode de desenvolupament del treball serà el que s'adapti a aquest model, en concret el mètode seguit a l'assignatura d'Enginyeria del Programari I, que està inspirat en el Rational Unified Process, també de l'OMG.

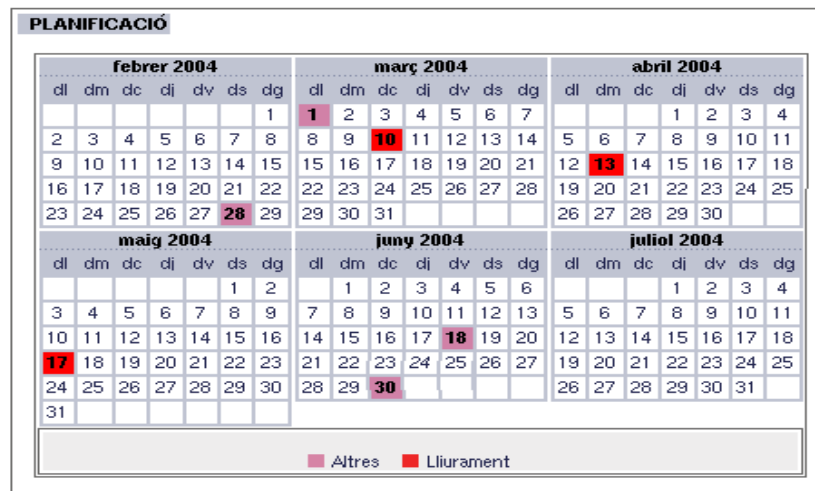


Figura 8.1: Calendari

8.1.4 Recursos

Primerament com es pot suposar treballa en una operadora de telecomunicacions dedicat a la gestió de la xarxa de fibra òptica. El traspàs cap a un sistema GIS és un projecte que estic liderant dins de l'empresa i necessito un plantejament formal dels requeriments més importants i les solucions a adoptar perquè aquesta migració es faci correctament. Actualment s'ha fet un traspàs parcial amb un model que s'adapta més a les dades que teníem que no pas a les que necessitem.

Com a recursos tècnics tinc intenció de fer servir sempre que pugui programari lliure, en el sentit de programari de codi font obert i lliure (<http://www.gnu.org>). Entre aquest programari hi ha:

- Documentació: OpenOffice.org (<http://www.openoffice.org>)
- Diagrames: Dia (<http://www.lysator.liu.se/alla/dia/dia.html>) ó
- Diagrames: Graphviz (<http://www.research.att.com/sw/tools/graphviz>)
- Sistema Gestor de Bases de Dades: PostgreSQL (<http://www.postgre.org>)
- Biblioteca d'enllaç SGBD↔GIS: PostGIS (<http://postgis.refractory.net>)
- Sistema GIS: Quantum GIS (<http://qgis.sourceforge.net>) ó
- Sistema GIS: JUMP (<http://www.vividsolutions.com/jump>)

També em proposo, en el cas que la codificació surti endavant, publicar aquesta sota una llicència lliure i editar-la en anglès, fins i tot la nomenclatura i vocabulari. També fer servir un repositori comú de codi de l'estil de SourceForge (<http://sourceforge.net>) o Savannah (<http://savannah.gnu.org>).

Capítol 9

Recursos de programari

Es descriuen a continuació els diferents aplicatius previstos per la implementació del codi amb un extracte de la informació que presenten a les diferents webs i a la documentació d'aquests programes.

9.1 PostgreSQL

PostgreSQL és la més avançada aplicació de base de dades de codi obert del món. Porta en desenvolupament des de 1986 i com avantatges més importants presenta els següents:

- Utilització ilimitada en tots els aspectes relacionats amb la llicència, fins i tot en les tasques de recerca i desenvolupament.
- Suport a l'usuari de primera línia, tant per la comunitat d'usuaris i desenvolupadors com er empreses especialitzades.
- Manteniment econòmic gràcies a un disseny que no necessita grans requeriments de gestió i afinament, tot i que es mantenen els compromisos de fiabilitat, estabilitat i rendiment.
- Programes de formació ajustats a les necessitats del món real, amb gran eficiència en el seu desenvolupament.
- Estabilitat legendària, amb exemples de companyies que el fan servir sense interrupcions des de fa anys i sense problemes d'estabilitat.
- Extensible gràcies a la possibilitat de manipular i particularitzar el codi font. Existeixen un gran nombre de desenvolupadors professionals i entusiàstes.
- Multiplataforma ja que és possible la seva execució nativa en 34 plataformes Unix i sobre **Windows** a través de la llibreria **Cywin** o de forma nativa amb desenvolupaments comercials.
- Capaç de gestionar grans quantitats de dades gràcies a la utilització de la tecnologia MVCC (Multiple row data storage strategy).
- Entorns d'usuari gràfics, desenvolupats en paral·lel tant per l'administració com pel disseny i desenvolupament de bases de dades.



Figura 9.1: Captura de la pàgina web de PostgreSQL

Igualment **PostgreSQL** té els següents avantatges tècnics:

- Acompleix els estàndards ACID i ANSI SQL.
- Integritat referencial.
- Replicació.
- Interfícies natives per ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python i Ruby.
- Regles, Vistes, Triggers, Seqüències, Herència, Outer Joins, Subsel·leccions.
- Admet Unicode.
- API obert.
- Procediments emmagatzemats i llenguatge procedimental PL/pgSQL.
- Suport natiu per SSL i Kerberos.
- Suport en calent (solucions comercials).
- Bloqueig a nivell superior a fila.
- Indexació funcional i parcial.
- Suporta UNION, UNION ALL i EXCEPT a les consultes.

- Càrrega d'extensions com MD5, SHA1, XML, ...
- Eines per la generació de SQL portable a altres sistemes.
- Tipatge de dades.
- Compatibilitat amb altres sistemes de bases de dades per facilitar la transició.

9.2 JUMP

La Java Unified Mapping Platform (**JUMP**) és una plataforma de codi obert per la visualització i edició de dades espacials. Inclou funcions espacials i geogràfiques. També està dissenyada per ser una bastida pel desenvolupament d'aplicacions de tractament de dades espacials.

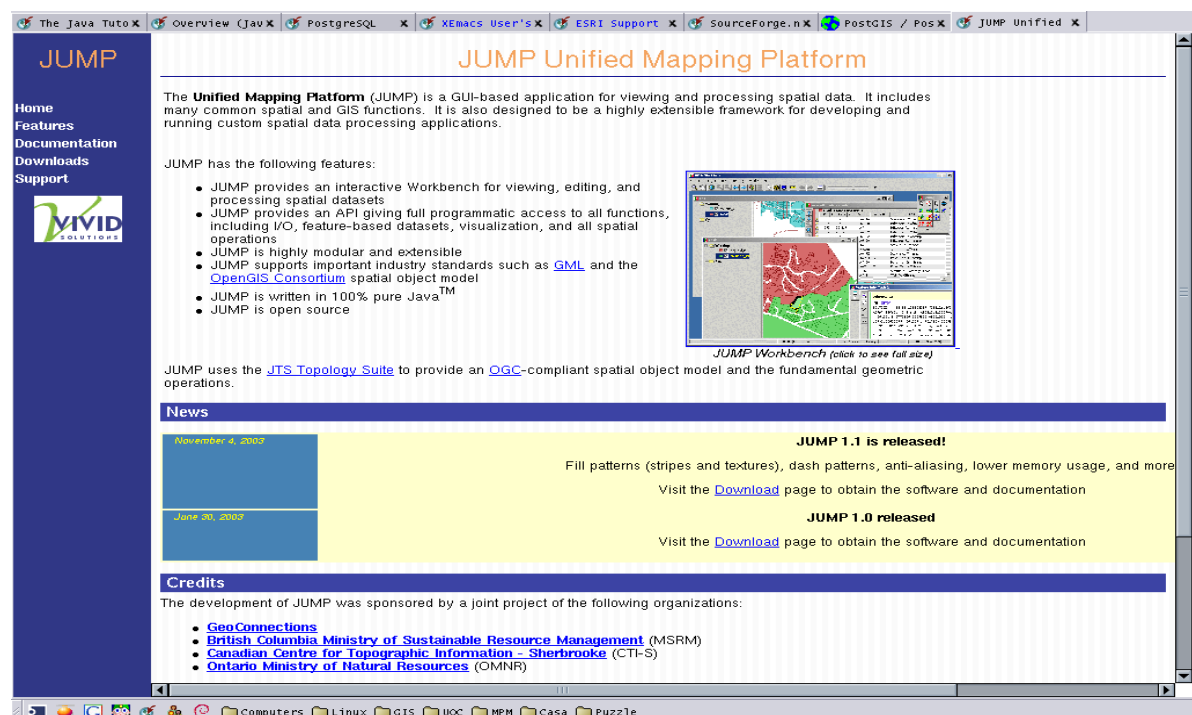


Figura 9.2: Captura de la pàgina web de JUMP

Entre les funcionalitats més importants destaquen:

- Entorn de treball interactiu per la visualització, edició i procesat d'informació espacial.
- Complerta API pel desenvolupament a baix nivell.
- Modular i altament extensible.
- Suporta els estàndards GML i **OpenGis Consortium**.

Ha estat desenvolupat per **Vivid Solutions** [24] amb **Refractions Research** [25] i esponsoritzat pels organismes oficials del Canadà **GeoConnections** Canadian Geospatial Data Infrastructure, **MSRM** British Columbia Ministry of Sustainable Resource Management, **CTI-S** Canadian Centre for Topographic Information - Sherbrooke i **OMNR** Ontario Ministry of Natural Resources.

Capítol 10

GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a

licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not **"Transparent"** is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, **"Title Page"** means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify

the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Capítol 11

Codi implementat

Es presenta el codi implementat fins el moment per part de l'autor. Com ja es comenta en aquest treball (Secció 1.1) es tracta d'una implementació que, en relació amb el treball, demostra la possibilitat real de connexió entre els diferents mòduls i programaris que es fan servir en el model. Aquesta possibilitat queda demostrada amb el codi desenvolupat.

Cal tenir present però que el model utilitzat en la implementació ha patit diferents modificacions respecte el que s'ha desenvolupat en aquest treball i que tal com ja s'exposava anteriorment tota la implementació s'ha fet en anglès i sota llicència de codi obert.

La implementació es pot considerar doncs com un projecte paral·lel a aquest treball i es pot trobar a internet [9].

11.1 Model de Dades

El node *Mode/DB* es desenvolupa en SQL i PL/pgSQL i crea una base de dades i les rutines de consulta avançades per la explotació d'aquestes dades. Es crea sobre **PostgreSQL** amb l'extensió geomètrica **PostGIS**.

```
#!/bin/sh
#####
# $Id$
#####
# loader.sh
# Loads all sql files to PostgreSQL server.
#####
# Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
#
# This file is part of TeNeM the Telecommunications Network Model.
#
# TeNeM is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
```

```

#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
#####

# First change at your convenience the variables.
# Second make loader.sh executable (chmod +x).
# Last call script from shell like:
# $ ./loader.sh

# Name of the database in postgres
DBNAME=TeNeM

# Name of standard user in postgres
USER=santiago

# Name of postgres superuser
PGSUPERUSER=postgres

# Some directories
PGDIR=/usr/local/postgresql/bin
POSTGISDIR=/usr/local/postgresql/share/contrib
DATADIR=/home/santiago/Documentos/dev/TeNeM/DBModel

echo -e "-- -- -- -- --"
echo -e "-- Cleaning database $DBNAME ..."
$PGDIR/dropdb $DBNAME

echo -e "\n-- Creating database $DBNAME ..."
$PGDIR/createdb $DBNAME

echo -e "\n-- Add language PL/pgSQL to $DBNAME."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/createlang plpgsql TeNeM

echo -e "\n-- Add postgis.sql to $DBNAME."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -q -f ' $POSTGISDIR/postgis.sql

echo -e "\n-- Add spatial_ref_sys.sql to $DBNAME."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -q -f ' $POSTGISDIR/spatial_ref_sys.sql

echo -e "\n-- Grant acces to geometric tables to $USER 1/4."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -c ' "'GRANT SELECT ON spatial_ref_sys TO" $USER ";"

echo -e "\n-- Grant acces to geometric tables to $USER 2/4."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -c ' "'GRANT SELECT ON geometry_columns TO" $USER ";"

echo -e "\n-- Grant acces to geometric tables to $USER 3/4."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -c ' "'GRANT INSERT ON geometry_columns TO" $USER ";"

echo -e "\n-- Grant acces to geometric tables to $USER 4/4."
echo -e "-- Insert user $PGSUPERUSER password:"
su $PGSUPERUSER -c $PGDIR/psql ' -d TeNeM -c ' "'GRANT UPDATE ON geometry_columns TO" $USER ";"

echo -e "\n-- Load data in $DBNAME ..."
$PGDIR/psql -d $DBNAME -q -f $DATADIR/TeNeM.sql
$PGDIR/psql -d $DBNAME -q -f $DATADIR/InsertData.sql
$PGDIR/psql -d $DBNAME -q -f $DATADIR/GeometricTeNeM.sql
$PGDIR/psql -d $DBNAME -q -f $DATADIR/GeometricInsertData.sql

echo -e "\n-- Load PL/pgSQL procedures ..."

```

```

$PGDIR/psql -d $DBNAME -q -f $DATADIR/InsertDataAutomatic.sql
$PGDIR/psql -d $DBNAME -q -f $DATADIR/CablePathFrom.sql

echo -e "\n-- Launch database client with $DBNAME ..."
echo -e "-- -- -- -- --"
$PGDIR/psql $DBNAME

-- $Id$
-----
-- TeNeM.sql
-- Creates a PostgreSQL database with a fiber optic network.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Create a db from shell. Then open this file from 'psql' with \i command:
--
-- $ createdb TeNeM
--
-- $ psql TeNeM
-- Welcome to psql, the PostgreSQL interactive terminal.
--
-- Type: \copyright for distribution terms
--       \h for help with SQL commands
--       \? for help on internal slash commands
--       \g or terminate with semicolon to execute query
--       \q to quit
--
-- TeNeM=> \i /path/to/document/TeNeM.sql
-----

-- Drop existing tables and sequences

-- Since loader.sh drops database this is not necessary -----
--DROP SEQUENCE Ids_seq;

--DROP TABLE FibersSplices CASCADE;
--DROP TABLE CablesInJoint CASCADE;
--DROP TABLE CablesJoints CASCADE;
--DROP TABLE EdgesInJoint CASCADE;
--DROP TABLE Joints CASCADE;

--DROP TABLE Fibers CASCADE;
--DROP TABLE Cables CASCADE;
--DROP TABLE Ducts CASCADE;
--DROP TABLE Edges CASCADE;

--DROP TABLE EdgeSubtypes CASCADE;
--DROP TABLE DuctSubtypes CASCADE;
--DROP TABLE CableSubtypes CASCADE;
--DROP TABLE Promoters CASCADE;
-- End modifications from loader.sh -----

```



```
-- Create sequence. Starts so far to work with sample data with no interference
-- with the primary keys of data inserted by hand.

CREATE SEQUENCE Ids_seq MINVALUE 1000000;

-- Create tables:

-- Create domains for fields in main tables.

BEGIN WORK;

CREATE TABLE EdgeSubtypes (
    Subtype varchar (10) PRIMARY KEY
);

CREATE TABLE DuctSubtypes (
    Subtype varchar (10) PRIMARY KEY
);

CREATE TABLE CableSubtypes (
    Subtype varchar (10) PRIMARY KEY
);

CREATE TABLE Promoters (
    Name varchar (64) PRIMARY KEY
);

COMMIT;

-- Create main tables.

BEGIN WORK;

CREATE TABLE Edges (
    IdEdge int DEFAULT nextval('Ids_seq') PRIMARY KEY,
    INF varchar(10) NOT NULL,
    Subtype varchar NOT NULL,
    Promoter varchar NOT NULL,

    FOREIGN KEY (Subtype) REFERENCES EdgeSubtypes ON UPDATE CASCADE,
    FOREIGN KEY (Promoter) REFERENCES Promoters ON UPDATE CASCADE
);

CREATE TABLE Ducts (
    IdDuct int DEFAULT nextval('Ids_seq') PRIMARY KEY,
    Edge int NOT NULL,
    Subtype varchar NOT NULL,

    FOREIGN KEY (Edge) REFERENCES Edges ON DELETE CASCADE,
    FOREIGN KEY (Subtype) REFERENCES DuctSubtypes ON UPDATE CASCADE
);

CREATE TABLE Cables (
    IdCable int DEFAULT nextval('Ids_seq') PRIMARY KEY,
    Duct int NOT NULL,
    Subtype varchar NOT NULL,
    NFibers int NOT NULL,

    FOREIGN KEY (Duct) REFERENCES Ducts ON DELETE CASCADE,
    FOREIGN KEY (Subtype) REFERENCES CableSubtypes ON UPDATE CASCADE,
    CHECK (NFibers >= 0)
);

CREATE TABLE Fibers (
    IdFiber int DEFAULT nextval('Ids_seq') PRIMARY KEY,
    Cable int NOT NULL,
```

```

    Crown int NOT NULL DEFAULT 0,
    Tube int NOT NULL,
    Fiber int NOT NULL,

    FOREIGN KEY (Cable) REFERENCES Cables ON DELETE CASCADE,
    CHECK (Crown >= 0),
    CHECK (Tube >= 0),
    CHECK (Fiber >= 0)
);

COMMIT;

-- Create connections between edges.

BEGIN WORK;

CREATE TABLE Joints (
    IdJoint int DEFAULT nextval('Ids_seq') PRIMARY KEY
);

CREATE TABLE EdgesInJoint (
    Joint int NOT NULL,
    Edge int NOT NULL,

    PRIMARY KEY (Joint, Edge),
    FOREIGN KEY (Joint) REFERENCES Joints ON DELETE CASCADE,
    FOREIGN KEY (Edge) REFERENCES Edges ON DELETE CASCADE
);

CREATE TABLE CablesJoints (
    IdCablesJoint int DEFAULT nextval('Ids_seq') PRIMARY KEY,
    Joint int NOT NULL,

    FOREIGN KEY (Joint) REFERENCES Joints ON DELETE CASCADE
);

CREATE TABLE CablesInJoint (
    CablesJoint int NOT NULL,
    Cable int NOT NULL,

    PRIMARY KEY (CablesJoint, Cable),
    FOREIGN KEY (CablesJoint) REFERENCES CablesJoints ON DELETE CASCADE,
    FOREIGN KEY (Cable) REFERENCES Cables ON DELETE CASCADE
);

CREATE TABLE FibersSplices (
    Fiber1 int NOT NULL,
    Fiber2 int NOT NULL,
    CablesJoint int NOT NULL,

    PRIMARY KEY (CablesJoint, Fiber1, Fiber2),
    FOREIGN KEY (Fiber1) REFERENCES Fibers ON DELETE CASCADE,
    FOREIGN KEY (Fiber2) REFERENCES Fibers ON DELETE CASCADE,
    FOREIGN KEY (CablesJoint) REFERENCES CablesJoints ON DELETE CASCADE
);

COMMIT;

-- $Id$
-----
-- Tables.sql
-- Shows table rows from TeNeM database.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--

```

```

-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Open this file from 'psql' whith \i command:
--
-- TeNeM=> \i /path/to/document/Tables.sql
-----

-- Show tables and sequences

SELECT * FROM Promoters;
SELECT * FROM CableSubtypes;
SELECT * FROM DuctSubtypes;
SELECT * FROM EdgeSubtypes;

SELECT * FROM Edges;
SELECT * FROM Ducts;
SELECT * FROM Cables;
SELECT * FROM Fibers;

SELECT * FROM Joints;
SELECT * FROM EdgesInJoint;
SELECT * FROM CablesJoints;
SELECT * FROM CablesInJoint;
SELECT * FROM FibersSplices;

SELECT * FROM Ids_seq;

-- $Id$
-----
-- InsertDataAutomatic.sql
-- Fills database with a fiber optic network.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Open this file from 'psql' whith \i command:
--
-- TeNeM=> \i /path/to/document/InsertDataAutomatic.sql
--
-- Sample of use: 'SELECT InsertCable(xxx);'

```

```

-----

-- Function int InsertCable(int)
-- Parameter int: Id of duct where create cable.
-- Returns: '0' if all is right.
-----

-- Drop function before create it.
DROP FUNCTION InsertCable(int);

CREATE OR REPLACE FUNCTION InsertCable(int)
  RETURNS int AS '
DECLARE
  idDuct1 ALIAS FOR $1;
BEGIN

  INSERT INTO Cables(Duct, Subtype, NFibers)
    VALUES(idDuct1, 'FlexTube', 288);

  RETURN 0;
END;
' LANGUAGE plpgsql;

-----

-- Function int InsertDucts(int, int)
-- Parameter int: Id of edge where create ducts.
-- Parameter int: Number of ducts created in edge.
-- Returns: '0' if all is right.
-----

-- Drop function before create it.
DROP FUNCTION InsertDucts(int, int);

CREATE OR REPLACE FUNCTION InsertDucts(int, int)
  RETURNS int AS '
DECLARE
  idEdge1 ALIAS FOR $1;
  n ALIAS FOR $2;
BEGIN

  IF n < 1 THEN
    RAISE EXCEPTION 'Parameter must be > 0';
    RETURN 1;
  END IF;

  FOR i IN 1..n LOOP
    INSERT INTO Ducts(Edge, Subtype) VALUES(idEdge1, '125mm');
  END LOOP;

  RETURN 0;
END;
' LANGUAGE plpgsql;

-----

-- Function int CreateEdges(int)
-- Parameter int: Number of rows created in table.
-- Returns: '0' if all is right.
-- This function creates 'n' edges
-----

-- Drop function before create it.
DROP FUNCTION CreateEdges(int);

CREATE OR REPLACE FUNCTION CreateEdges(int)
  RETURNS int AS '

```

```

DECLARE
    n ALIAS FOR $1;
    idEdge1 int;
    idEdge2 int;
    idJoint1 int;
BEGIN
    IF n < 1 THEN
        RAISE EXCEPTION 'Parameter must be > 0';
        RETURN 1;
    END IF;

    FOR i IN 1..n LOOP

        INSERT INTO Joints DEFAULT VALUES;

        SELECT INTO idJoint1 max(IdJoint) FROM Joints;

        SELECT INTO idEdge1 max(IdEdge) FROM Edges;

        INSERT INTO Edges(INF, Subtype, Promoter)
            VALUES('INF04-0099', 'Trench', 'BEROTELECOM');

        SELECT INTO idEdge2 max(IdEdge) FROM Edges;

        INSERT INTO EdgesInJoint(Joint, Edge)
            VALUES(idJoint1, idEdge1);
        INSERT INTO EdgesInJoint(Joint, Edge)
            VALUES(idJoint1, idEdge2);

        PERFORM InsertDucts(idEdge2, 2);

    END LOOP;
    RETURN 0;
END;
' LANGUAGE plpgsql;

-----
-- Function int CreateNetwork(int)
-- Parameter int: Number of edges in the network.
-- Returns: '0' if all is right.
-- This function creates 'n' edges jointed and one cable also jointed along
-- the network.
-----

-- Drop function before create it.
DROP FUNCTION CreateNetwork(int);

CREATE OR REPLACE FUNCTION CreateNetwork(int)
    RETURNS int AS '
DECLARE
    n ALIAS FOR $1;
    idJoint0 int;
    idEdge1 int;
    idEdge2 int;
    idDuct1 int;
    idDuct2 int;
    idCablesJoints0 int;
    idCable1 int;
    idCable2 int;
BEGIN
    IF n < 1 THEN
        RAISE EXCEPTION 'Parameter must be > 0';
        RETURN 1;
    END IF;

```

```

INSERT INTO Edges(INF, Subtype, Promoter)
VALUES('INF04-0055', 'Trench', 'BEROTELECOM');

SELECT INTO idEdge1 max(IdEdge) FROM Edges;

INSERT INTO Ducts(Edge, Subtype) VALUES(idEdge1, '125mm');
INSERT INTO Ducts(Edge, Subtype) VALUES(idEdge1, '125mm');

SELECT INTO idDuct1 max(IdDuct) FROM Ducts WHERE Edge = idEdge1;

INSERT INTO Cables(Duct, Subtype, NFibers)
VALUES(idDuct1, 'FlexTube', 288);

FOR i IN 1..n LOOP

    INSERT INTO Joints DEFAULT VALUES;

    SELECT INTO idJoint0 max(IdJoint) FROM Joints;

    INSERT INTO Edges(INF, Subtype, Promoter)
    VALUES('INF04-0044', 'Trench', 'BEROTELECOM');

    SELECT INTO idEdge2 max(IdEdge) FROM Edges;

    INSERT INTO EdgesInJoint(Joint, Edge)
    VALUES(idJoint0, idEdge1);
    INSERT INTO EdgesInJoint(Joint, Edge)
    VALUES(idJoint0, idEdge2);

    INSERT INTO Ducts(Edge, Subtype) VALUES(idEdge2, '125mm');
    INSERT INTO Ducts(Edge, Subtype) VALUES(idEdge2, '125mm');

    SELECT INTO idDuct2 max(IdDuct) FROM Ducts WHERE Edge = idEdge2;

    INSERT INTO Cables(Duct, Subtype, NFibers)
    VALUES(idDuct2, 'FlexTube', 288);

    INSERT INTO CablesJoints(Joint) VALUES(idJoint0);

    SELECT INTO idCablesJoints0 max(IdCablesJoint) FROM CablesJoints;
    SELECT INTO idCable1 max(IdCable) FROM Cables WHERE Duct = idDuct1;
    SELECT INTO idCable2 max(IdCable) FROM Cables WHERE Duct = idDuct2;

    INSERT INTO CablesInJoint(CablesJoint, Cable)
    VALUES (idCablesJoints0, idCable1);
    INSERT INTO CablesInJoint(CablesJoint, Cable)
    VALUES (idCablesJoints0, idCable2);

    idEdge1 := idEdge2;
    idDuct1 := idDuct2;

END LOOP;
RETURN 0;
END;
' LANGUAGE plpgsql;

-- $Id$
-----
-- InsertData.sql
-- Fills database with a fiber optic network.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by

```

```

-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Open this file from 'psql' whith \i command:
--
-- TeNeM=> \i /path/to/document/InsertData.sql
-----

-- Insert data

BEGIN WORK;

INSERT INTO Promoters(Name) VALUES('BEROTELECOM');

COMMIT;

BEGIN WORK;

INSERT INTO CableSubtypes(Subtype) VALUES('Loose Tube');
INSERT INTO CableSubtypes(Subtype) VALUES('FlexTube');
INSERT INTO CableSubtypes(Subtype) VALUES('Ribbon');

COMMIT;

BEGIN WORK;

INSERT INTO DuctSubtypes(Subtype) VALUES('125mm');
INSERT INTO DuctSubtypes(Subtype) VALUES('40mm');
INSERT INTO DuctSubtypes(Subtype) VALUES('160mm');

COMMIT;

BEGIN WORK;

INSERT INTO EdgeSubtypes(Subtype) VALUES('Trench');
INSERT INTO EdgeSubtypes(Subtype) VALUES('Aerial');
INSERT INTO EdgeSubtypes(Subtype) VALUES('Tunnel');

COMMIT;

BEGIN WORK;

INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(1, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(2, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(3, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(4, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(5, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(6, 'INF04-0001', 'Trench', 'BEROTELECOM');
INSERT INTO Edges(IdEdge, INF, Subtype, Promoter) VALUES(7, 'INF04-0001', 'Trench', 'BEROTELECOM');

COMMIT;

BEGIN WORK;

INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(201, 1, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(202, 1, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(203, 2, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(204, 2, '125mm');

```

```

INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(205, 3, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(206, 3, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(207, 4, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(208, 4, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(209, 5, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(210, 5, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(211, 6, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(212, 6, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(213, 7, '125mm');
INSERT INTO Ducts(IdDuct, Edge, Subtype) VALUES(214, 7, '125mm');

COMMIT;

BEGIN WORK;

INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(301, 201, 'Ribbon', 96);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(302, 203, 'Ribbon', 96);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(303, 205, 'Ribbon', 96);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(304, 201, 'Ribbon', 144);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(305, 207, 'Ribbon', 144);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(306, 209, 'Ribbon', 144);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(307, 213, 'Ribbon', 144);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(308, 211, 'FlexTube', 16);
INSERT INTO Cables(IdCable, Duct, Subtype, NFibers) VALUES(309, 205, 'FlexTube', 16);

COMMIT;

BEGIN WORK;

INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 1);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 2);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 3);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 4);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 5);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 6);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 7);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(301, 0, 2, 8);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 1);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 2);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 3);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 4);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 5);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 6);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 7);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(302, 0, 2, 8);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 1);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 2);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 3);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 4);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 5);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 6);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 7);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(303, 0, 2, 8);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 1);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 2);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 3);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 4);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 5);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 6);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 7);
INSERT INTO Fibers(Cable, Crown, Tube, Fiber) VALUES(304, 0, 1, 8);

COMMIT;

BEGIN WORK;

INSERT INTO Joints(IdJoint) VALUES (101);

```



```
INSERT INTO Joints(IdJoint) VALUES (102);
INSERT INTO Joints(IdJoint) VALUES (103);
INSERT INTO Joints(IdJoint) VALUES (104);
INSERT INTO Joints(IdJoint) VALUES (105);
INSERT INTO Joints(IdJoint) VALUES (106);
INSERT INTO Joints(IdJoint) VALUES (107);

COMMIT;

BEGIN WORK;

INSERT INTO EdgesInJoint(Joint, Edge) VALUES(101, 1);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(102, 1);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(102, 2);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(102, 4);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(103, 2);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(103, 3);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(103, 6);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(104, 3);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(105, 4);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(105, 5);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(106, 5);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(106, 6);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(106, 7);
INSERT INTO EdgesInJoint(Joint, Edge) VALUES(107, 7);

COMMIT;

BEGIN WORK;

INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(401, 101);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(402, 102);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(403, 103);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(404, 104);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(405, 101);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(406, 102);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(407, 105);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(408, 106);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(409, 107);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(410, 103);
INSERT INTO CablesJoints(IdCablesJoint, Joint) VALUES(411, 104);

COMMIT;

BEGIN WORK;

INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(401, 301);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(402, 301);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(402, 302);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(403, 302);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(403, 303);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(404, 303);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(405, 304);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(406, 304);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(406, 305);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(407, 305);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(407, 306);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(408, 306);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(408, 307);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(408, 308);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(409, 307);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(410, 308);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(410, 309);
INSERT INTO CablesInJoint(CablesJoint, Cable) VALUES(411, 309);

COMMIT;
```

```
-- $Id$
-----
-- GeometricTeNeM.sql
-- Creates the geometric structure of TeNeM.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Add PostGIS functionality to database. It is necessary to modify some
-- privileges: postgis.sql defines C functions so you need USAGE privileges
-- on this language, but is better (I think) to call these scripts from
-- superuser and change visibility of tables generated. Then load this file
-- from 'psql' with \i command:
--
-- $ createdb TeNeM
-- # createlang plpgsql TeNeM
-- # psql -d TeNeM -f /path/to/document/postgis.sql
-- # psql -d TeNeM -f /path/to/document/spatial_ref_sys.sql
-- # psql -d TeNeM -c 'GRANT SELECT ON spatial_ref_sys TO username;'
-- # psql -d TeNeM -c 'GRANT SELECT ON geometry_columns TO username;'
-- # psql -d TeNeM -c 'GRANT INSERT ON geometry_columns TO username;'
-- # psql -d TeNeM -c 'GRANT UPDATE ON geometry_columns TO username;'
-- $ psql TeNeM
-- Welcome to psql, the PostgreSQL interactive terminal.
--
-- Type: \copyright for distribution terms
--       \h for help with SQL commands
--       \? for help on internal slash commands
--       \g or terminate with semicolon to execute query
--       \q to quit
--
-- TeNeM=> \i /path/to/document/GeometricTeNeM.sql
-----

-- Create geometric attributes on tables:

BEGIN WORK;

-- Table Edges is the way to connect the network model to the 'real' world
-- in the GIS representation. The table is defined with the non geometric
-- attributes and then the geometry is added with a PostGIS function.
-- It is a similar way to Joints.
-- SRID (Spatial Reference ID) is '0' for compatibility with Jump 1.1.1

SELECT AddGeometryColumn('public','edges','centerline','0','LINESTRING','2');

COMMIT;

BEGIN WORK;

-- Table Joints is the way to connect the network model to the 'real' world
-- in the GIS representation. The table is defined with the non geometric
```

```

-- attributes and then the geometry is added with a PostGIS function.
-- It is a similar way to Edges.
-- SRID (Spatial Reference ID) is '0' for compatibility with Jump 1.1.1

SELECT AddGeometryColumn('public','joints','position','0','POINT','2');

COMMIT;

-- $Id$
-----
-- GeometricInsertData.sql
-- Fills database with a geometric fiber optic network.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Open this file from 'psql' whith \i command:
--
-- TeNeM=> \i /path/to/document/GeometricInsertData.sql
-----

-- Insert data in geometric columns:

BEGIN WORK;

UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 18, 10 21, 16 23, 28 86, 44 91 )',
'0') WHERE IdEdge = 1;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 28, 10 31, 16 23, 28 76, 44 81 )',
'0') WHERE IdEdge = 2;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 38, 10 41, 16 23, 28 66, 44 71 )',
'0') WHERE IdEdge = 3;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 48, 10 51, 16 23, 28 56, 44 61 )',
'0') WHERE IdEdge = 4;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 58, 10 61, 16 23, 28 46, 44 51 )',
'0') WHERE IdEdge = 5;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 68, 10 71, 16 23, 28 36, 44 41 )',
'0') WHERE IdEdge = 6;
UPDATE Edges SET Centerline = LineStringFromText('LINESTRING( 0 78, 10 81, 16 23, 28 26, 44 31 )',
'0') WHERE IdEdge = 7;

COMMIT;

BEGIN WORK;

UPDATE Joints SET Position = PointFromText('POINT( 10 10)', '0') WHERE IdJoint = 101;
UPDATE Joints SET Position = PointFromText('POINT( 20 10)', '0') WHERE IdJoint = 102;
UPDATE Joints SET Position = PointFromText('POINT( 30 10)', '0') WHERE IdJoint = 103;
UPDATE Joints SET Position = PointFromText('POINT( 40 10)', '0') WHERE IdJoint = 104;
UPDATE Joints SET Position = PointFromText('POINT( 50 10)', '0') WHERE IdJoint = 105;
UPDATE Joints SET Position = PointFromText('POINT( 60 10)', '0') WHERE IdJoint = 106;
UPDATE Joints SET Position = PointFromText('POINT( 70 10)', '0') WHERE IdJoint = 107;

```

```

COMMIT;

-- $Id: CablePathFrom.sql,v 1.5 2004/04/29 23:36:06 santiagopm Exp $
-----
-- CablePathFrom.sql
-- Shows cables that connect topologically to Cable from a CablesConnector.
-----
-- Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
--
-- This file is part of TeNeM the Telecommunications Network Model.
--
-- TeNeM is free software; you can redistribute it and/or modify
-- it under the terms of the GNU General Public License as published by
-- the Free Software Foundation; either version 2 of the License, or
-- (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-----
-- Open this file from 'psql' whith \i command:
--
-- TeNeM=> \i /path/to/document/CablePathFrom.sql
-----

-- Drop tables and functions (necessary if changing return type).

DROP FUNCTION CablePathFrom(int, int);

-- Function CablePathFrom(IdCable int, IdJoint int):CablesInJoint
-- Parameter IdCable: Cable from where searching new connected cables.
-- Parameter IdCablesJoint: Joint from where the cable starts. If '0' there
-- is not any starting CablesJoint so the function reaches normally the two
-- cable's joints (two directions).
-- Returns: A set of CablesInJoint with cables that connects topologically
-- to the original Cable parameter. The direction of the search is established
-- by the joint in parameter.
-- Sample of use: 'SELECT * FROM CablePathFrom(xxx, yyy);'

CREATE OR REPLACE FUNCTION CablePathFrom(int, int)
  RETURNS SETOF CablesInJoint AS '
DECLARE
  IdCable ALIAS FOR $1;
  IdCablesJoint ALIAS FOR $2;
  joint1 CablesInJoint%ROWTYPE;
  joint2 CablesInJoint%ROWTYPE;
  joint3 CablesInJoint%ROWTYPE;
BEGIN

  -- Select CablesJoints that connects this cable with others, it must
  -- be two or rarely one CablesJoints by Cable, so select the one
  -- that is on the opposite tip of the cable.
  FOR joint1 IN SELECT * FROM CablesInJoint
  WHERE Cable = IdCable
  AND CablesJoint <> IdCablesJoint LOOP

    -- Select all the cables that starts in this CablesJoint, so
    -- we can follow the paths from the original cable.
    FOR joint2 IN SELECT * FROM CablesInJoint
    WHERE CablesJoint = joint1.CablesJoint
    AND Cable <> IdCable LOOP

```

```

-- Now from the cable that connects the other tip of the original
-- cable we searches new cables connected, calling recursively
-- the same function.
FOR joint3 IN SELECT * FROM
CablePathFrom( joint2.Cable, joint2.CablesJoint) LOOP

    RETURN NEXT joint3;

END LOOP; -- joint3

RETURN NEXT joint2;

END LOOP; -- joint2

END LOOP; -- joint1

-- Always required.
RETURN;
END;
' LANGUAGE plpgsql;

```

11.2 Client GIS

El node *ClientGIS* connecta des d'un client de **JUMP** amb les dades del model de manera que fa servir aquest aplicatiu com interfície gràfica per l'edició de la xarxa de telecomunicacions.

```

/*****
* $Id$
*
* Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
*
* This file is part of TeNeM the Telecommunications Network Model.
*
* TeNeM is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*****/
package tenem;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashMap;
import java.util.List;

import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JLabel;

```

```

import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JRootPane;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;

import net.refractions.postgis.PostGISDataSource;
import net.refractions.postgis.PostGISSaveDataSourceQueryChooser;

import com.vividsolutions.jump.feature.Feature;
import com.vividsolutions.jump.workbench.WorkbenchContext;
import com.vividsolutions.jump.workbench.datasource.DataSourceQueryChooserManager;
import com.vividsolutions.jump.workbench.ui.WorkbenchFrame;

/**
 *
 */
public class TeNeMClientDialog extends JDialog implements ActionListener {

    static final String TENEM_GET_CABLES = "Cables";
    static final String TENEM_DRAW_CABLE = "Draw";
    static final String TENEM_OK = "OK";

    private WorkbenchContext context;

    private JPanel mainPanel;
    private JPanel btnPanel;
    private JLabel label;
    private JButton btnGetCables;
    private JButton btnDrawCable;
    private JButton btnOK;
    private DefaultListModel listModel;
    private JList list;
    private JScrollPane listScroller;

    public TeNeMClientDialog(final WorkbenchFrame frame){
        super(frame, "TeNeM", false);
        this.context = frame.getContext();

        mainPanel = new JPanel();
        btnPanel = new JPanel();
        label = new JLabel("TeNeM approaching!");
        btnGetCables = new JButton(TENEM_GET_CABLES);
        btnDrawCable = new JButton(TENEM_DRAW_CABLE);
        btnOK = new JButton(TENEM_OK);
        listModel = new DefaultListModel();
        list = new JList(listModel);
        listScroller = new JScrollPane(list);

        list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        list.setLayoutOrientation(JList.VERTICAL);
        list.setVisibleRowCount(-1);

        listScroller.setPreferredSize(new Dimension(200, 80));

        btnPanel.setLayout(new BoxLayout(btnPanel, BoxLayout.Y_AXIS));
        btnPanel.add(btnGetCables);
        btnPanel.add(Box.createVerticalGlue());
        btnPanel.add(btnDrawCable);
        btnPanel.add(Box.createVerticalGlue());
        btnPanel.add(btnOK);

        mainPanel.setLayout(new BorderLayout());
        mainPanel.add(label, BorderLayout.PAGE_START);
        mainPanel.add(listScroller, BorderLayout.EAST);
        mainPanel.add(btnPanel, BorderLayout.WEST);
    }

```

```

        getContentPane().add(mainPanel);

        btnGetCables.setActionCommand(TENEM_GET_CABLES);
        btnDrawCable.setActionCommand(TENEM_DRAW_CABLE);
        btnOK.setActionCommand(TENEM_OK);
        btnGetCables.addActionListener(this);
        btnDrawCable.addActionListener(this);
        btnOK.addActionListener(this);

        setUndecorated(true);
        setResizable(false);
        getRootPane().setWindowDecorationStyle(JRootPane.FRAME);
        getRootPane().setDefaultButton(btnGetCables);

        pack();
        setLocationRelativeTo(null);
    }

    public void actionPerformed(ActionEvent e) {
        if(TENEM_GET_CABLES.equals(e.getActionCommand())){
            System.out.println(TENEM_GET_CABLES);
            Object o = new Object();
            o = context
                .getLayerViewPanel()
                .getSelectionManager()
                .getFeatureSelection()
                .getFeaturesWithSelectedItems()
                .iterator()
                .next();
            //System.out.println(o.toString());
            listModel.addElement("Object: " + o.toString());
            Feature g = (Feature) o;
            //System.out.println(g.getID());
            listModel.addElement("IDEdge: " + g.getAttribute("idedge").toString());
            listModel.addElement("FID: " + new Integer(g.getID()).toString());

            List l = DataSourceQueryChooserManager.get(context.getBlackboard())
                .getSaveDataSourceQueryChoosers();
            TeNeMConnection tConn = new TeNeMConnection();
            Object u = new Object();
            u = l.get(5);
            int i = l.indexOf(u);
            PostGISDataSource pData = new PostGISDataSource();
            PostGISSaveDataSourceQueryChooser pSave = new PostGISSaveDataSourceQueryChooser(pData);
            //pData = (PostGISDataSource)l.get(5);
            System.out.println(l.get(5).toString());
            System.out.println(l.get(5).getClass().getName());
            pSave = (PostGISSaveDataSourceQueryChooser) u;
            HashMap prop = pSave.getProperties();
            int j = l.indexOf(pSave);
            int k = l.indexOf(pData);
            tConn.executeQuery("SELECT * FROM Cables");
        } else if (TENEM_OK.equals(e.getActionCommand())){
            this.hide();
        }
    }
}

/*****
 * $Id$
 *
 * Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
 *
 * This file is part of TeNeM the Telecommunications Network Model.
 *
 * TeNeM is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by

```

```

* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*****/
package tenem;

import com.vividsolutions.jump.workbench.plugin.Extension;
import com.vividsolutions.jump.workbench.plugin.PlugInContext;

/**
 * TeNeMConfiguration
 */
public class TeNeMConfiguration extends Extension {

    public void configure(PlugInContext context) throws Exception {
        new TeNeMPlugIn().initialize(context);
    }

    public String getName() {
        return ("Telecommunications Network Model <sourceforge.net/projects/tenem>");
    }

    public String getVersion() {
        return ("0.1");
    }
}

/*****
 * $Id$
 *
 * Copyright (c) 2004 Santiago Paya Miralta <santiagopm:wanadoo.es>
 *
 * This file is part of TeNeM the Telecommunications Network Model.
 *
 * TeNeM is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*****/
package tenem;

import com.vividsolutions.jump.workbench.plugin.AbstractPlugIn;
import com.vividsolutions.jump.workbench.plugin.PlugInContext;

/**
 * TeNeMPlugIn
 */
public class TeNeMPlugIn extends AbstractPlugIn {

```



```
/* (non-Javadoc)
 *
 */
public void initialize(PlugInContext context) throws Exception {
    context.getFeatureInstaller().addMainMenuItem(
        this,
        new String[] { "Tools", "TeNeM" },
        getName(),
        false,
        null,
        null);
}

/* (non-Javadoc)
 *
 */
public boolean execute(PlugInContext context) throws Exception {
//    context.getWorkbenchFrame().getOutputFrame().createNewDocument();
//    context.getWorkbenchFrame().getOutputFrame().addText("TeNeM is near!");
//    context.getWorkbenchFrame().getOutputFrame().surface();

    TeNeMClientDialog tenemClient = new TeNeMClientDialog(context.getWorkbenchFrame());
    tenemClient.setVisible(true);

    return true;
}

/* (non-Javadoc)
 *
 */
public String getName() {
    return ("TeNeM Client");
}
}
```