

# **INTEGRACIÓ DE GOOGLMAPS I QÜESTIONARIS IMS QTI**

PROJECTE FINAL DE CARRERA

Autor: José Bouzo Ruiz  
Directors: Toni Navarrete i Josep Blat  
Enginyeria en Informàtica  
Universitat Pompeu Fabra



*Vull dedicar aquest projecte en especial a la meua parella i als meus pares que han estat recolzant-me en tot moment.*

*Estic molt agraït també als meus tutors de projecte Toni Navarrete, per la seva amabilitat, disponibilitat, consells i ajuda; tanmateix per proposar l'article al workshop de TENCompetence, i a Josep Blat per la motivació que hem va donar per encetar el que ara és una realitat.*

*No hem puc oblidar del meu company José Luis Santos, sense ell tot això no hagués estat possible... També un record per a totes aquelles persones que no he esmentat específicament i que m'han donat la seva ajuda desinteressada durant tot el període universitari.*

*Gràcies a tots!*

D'aquest projecte s'ha creat un article amb el títol *Enhancing IMS QTI assessment with web maps* que ha estat acceptat i presentat al TENCompetence OpenWorkshop on Current research on IMS Learning Design and Lifelong Competence Development Infrastructures (Barcelona, Juny 2007).



## Resum

La utilització de mapes a l'ensenyament de la geografia esdevé de capital importància. En concret, els mapes també són sovint utilitzats en el procés d'avaluació dels alumnes. En aquest projecte es proposa una aproximació que desenvolupa un middleware per a connectar un motor que genera qüestionaris i preguntes definits segons l'especificació IMS QTI (APIS), amb un servei que ens proporciona mapes (GoogleMaps). La finalitat és aconseguir que treballin de manera conjunta, introduint els mínims canvis al motor. S'han modelat algunes interaccions contemplades a l'especificació, a més d'obrir noves formes d'interacció més naturals i intuïtives. L'aproximació és genèrica, i es podria fàcilment estendre per a altres serveis externs.

## Resumen

La utilización de mapas en el estudio de la geografía tiene especial relevancia. Concretamente, los mapas también son habitualmente utilizados en el proceso de evaluación de los alumnos. En este proyecto se propone una aproximación que desarrolla un middleware para conectar un motor que genera cuestionarios y preguntas definidas según la especificación IMS QTI (APIS), con un servicio que nos proporciona mapas (GoogleMaps). El objetivo es que trabajen de manera conjunta, introduciendo los mínimos cambios en el motor. Se han modelado algunas interacciones contempladas en la especificación, además de abrir nuevas maneras de interactuar más naturales e intuitivas. La aproximación es genérica y se podría fácilmente extender a otros servicios externos.

## Abstract

The maps in teaching geography environment are very important. Concretely, the maps are often used in the student evaluation process. Our approach develops a middleware to connect a engine that generates questionnaires and questions according to the specification IMS QTI (APIS), with a map server (GoogleMaps). The objective is that the two applications work together, introducing few changes in the engine. We have shaped some QTI interactions to the map and have opened new ways of interaction more natural and intuitive. This is a generic approximation and it's be able to be extended to other external services.



# Índex

---

<b>1. Introducció</b>	<b>9</b>
1.1 El context	9
1.2 Els objectius	10
1.3 Planificació del Projecte	11
1.4 Estructura de la memòria	14
<b>ESTAT DE L'ART</b>	
<b>2. L'e-Learning</b>	<b>17</b>
<b>3. Adaptació als estàndards</b>	<b>19</b>
3.1 Motivació per a generar estàndards	19
3.2 El procés d'estandardització	20
3.3 Els estàndards a l'e-Learning	21
3.4 IMS Global Learning Consortium	22
<b>4. IMS Question &amp; Test Interoperability</b>	<b>23</b>
4.1 Motivació	23
4.2 Evolució de l'especificació i situació actual	24
4.3 Característiques tècniques de l'especificació	25
4.4 Les preguntes	26
4.5 El nucli	27
4.6 Les interaccions	27
4.7 Els qüestionaris	35
4.8 Taxonomia	36
<b>5. APIS</b>	<b>37</b>
5.1 Definició del Projecte	37
5.2 Estructura	37
5.3 Implementació	38
5.4 Estat Actual	39
<b>6. GoogleMaps</b>	<b>41</b>
6.1 Introducció	41
6.2 Característiques i funcionament del servei	41
6.3 JavaScript	46
6.4 GoogleMaps com a eina d'aprenentatge	46
6.5 L'API de GoogleMaps	46
<b>DESENVOLUPAMENT DEL PROJECTE</b>	
<b>7. Anàlisi de Requeriments</b>	<b>51</b>
7.1 Diagrama de Casos d'ús	52
7.2 Requeriments funcionals	53
7.3 Requeriments no funcionals	54
7.4 Requeriments d'usuari	55
<b>8. Disseny i implementació del Projecte</b>	<b>57</b>
8.1 El Middleware	57
8.2 Representació de les dades de GoogleMaps	64
8.3 Estructura de dades del Middleware	67

<b>9. Les Interaccions i el processat de les respostes</b>	<b>73</b>
<b>9.1 Interacció com a imatge</b>	<b>73</b>
<b>9.2 Interaccions sobre el mapa</b>	<b>75</b>
<b>9.2.1 Choice</b>	<b>75</b>
<b>9.2.2 Order</b>	<b>77</b>
<b>9.2.3 PointIntoPolygon</b>	<b>78</b>
<b>9.2.4 LineIntoArea</b>	<b>81</b>
<b>9.2.5 DraggingMarkers</b>	<b>84</b>
<b>9.3 Complementant les interaccions (informació addicional)</b>	<b>86</b>
<b>10. Conclusions i treball futur</b>	<b>91</b>
<b>11. Bibliografia</b>	<b>95</b>
<b>ANNEXES</b>	<b>97</b>
<b>Glossari</b>	<b>99</b>
<i>Article</i>	
<b><i>Enhancing IMS QTI assessment with web maps</i></b>	<b>101</b>
<b>Joc de Proves</b>	<b>111</b>



# 1. Introducció

---

## 1.1 El Context

Actualment hi ha un gran interès en les aplicacions educatives que utilitzen Internet com a plataforma de distribució. Degut a això s'estan estudiant les possibilitats que ofereix la xarxa i les noves tecnologies per a respondre a la necessitat d'una educació personalitzada i no necessàriament presencial.

A partir de la segona meitat del s.XX van començar a posar-se en marxa (i esdevenir demanda) els cursos a distància de coneixements complementaris als bàsics amb l'objectiu de fer compatibles els estudis amb la feina o la família.

En aquest marc es situa l'e-Learning, basat en els següents punts:

- L'aprenentatge és assistit per les tecnologies de la informació, per tant, fomenta l'ús intensiu d'aquestes.
- Els continguts educatius han de ser més fàcils de crear, adoptar, intercanviar i distribuir.
- L'aprenentatge s'adapta al ritme de l'alumne.
- Les eines d'aprenentatge són disponibles independentment de les limitacions horàries o geogràfiques.
- L'intercanvi d'opinions i aportacions a través de les TIC's per part de l'alumnat i els docents, afavorint així la transferència, compartició i comunicació de coneixements creant una comunitat humana virtual ha de ser incentivada i potenciada d'una manera dinàmica i natural.

S'estan fent esforços comuns per a desenvolupar eines amb finalitats educatives ja que es considera que les noves tecnologies obren una sèrie de possibilitats que abans de la consolidació d'Internet no hi eren en l'àmbit de l'educació. Per tal d'unificar tot aquest desenvolupament es creen estàndards comuns amb la finalitat de fer una bona definició de requeriments bàsics d'aquest nou software i facilitar la existència de diferents productes; però sempre sota un mateix estàndard.

Hi ha diferents projectes educatius sota diverses entitats amb una clara vocació d'implantar un estàndard en aquesta àrea. Una d'aquestes entitats és IMS Global Learning Consortium que publica especificacions adreçades a diversos àmbits de l'aprenentatge.

Una de les parts més importants en el procés d'aprenentatge és l'avaluació dels coneixements adquirits, d'aquesta manera es certifica si realment l'alumne a consolidat els conceptes impartits. Aquesta fase tan important es fa mitjançant la resolució de qüestionaris i preguntes. Aquests qüestionaris i preguntes poden ser configurats i dissenyats de maneres molt diverses (tipus test, resposta oberta, completar frases, etc.) depenent de quina matèria s'està impartint i quin és l'objectiu en el qual es vol incidir. Amb tot això sorgeix també la necessitat d'una estandardització dels tipus de qüestionaris i preguntes que es poden construir i els tipus d'interaccions amb les quals pot respondre l'alumne.

L'especificació IMS Question & Test Interoperability descriu de manera unívoca l'estructura principal d'un qüestionari, compostat per diferents preguntes que es resolen d'una determinada manera (interacció).

En aquest projecte ens centrarem en l'àmbit de la Geografia i l'aprenentatge amb mapes, ja que l'estudi d'aquesta temàtica utilitzant aquesta eina té especial importància tant en l'educació bàsica com en l'adquisició de coneixements més avançats.

Més detalladament, tot estudi relacionat amb la Geografia necessita d'una representació conceptual de la Terra, aquesta representació ens la dona la Cartografia.

Anys enrere, a qualsevol aula on s'aprenia Geografia necessitava de mapes de diferents tipus, diferents escales, diferents informacions (polítics, físics, demogràfics,...). Tant és així que qualsevol docent per a poder impartir aquestes assignatures havia de tenir una quantitat de mapes elevada per a poder cobrir tot el temari que es volia impartir. En aquest punt ens sorgeix la necessitat d'utilitzar un software que ens proporcioni mapes amb una qualitat acceptable i de fàcil utilització. Aquí és on situem GoogleMaps, una tecnologia de caràcter gratuït oferta per Google de fàcil accés i molt intuïtiu.

Una altra part important és com avaluar la Geografia a través dels mapes, per tant cal generar un determinat tipus de qüestionaris utilitzant l'especificació abans esmentada incorporant-hi en la mateixa pregunta algun servei que ens proporcioni mapes d'una manera còmoda i flexible.

Resumint aquest projecte enllaçarà una manera de generar preguntes estàndard amb mapes construint una plataforma de comunicació entre el servei GoogleMaps i el motor que genera les preguntes per a que treballin conjuntament.

## 1.2 Objectius

En primer lloc, el projecte s'endinsa i centra el seu àmbit de desenvolupament en IMS Question and Test Interoperability (QTI). Aquesta especificació té com a objectiu descriure una estructura per a la representació de preguntes, qüestionaris, els seus corresponents resultats, avaluacions i altres conceptes relacionats com el feedback.

Aquesta especificació un cop definida, cal implementar-la, hi ha diversos projectes que intenten desenvolupar aquesta especificació; un d'aquest projectes *open-source* és APIS (Assessment Provision through Interoperable Segments).

APIS implementa una petita part de la versió 2.0 de l'especificació IMS QTI i en el Projecte Final de Carrera d'Helena Batlle [28] es va cobrir una gran part del que quedava de la versió 2.0 (centrada en els ítems) i una part important de la versió 2.1 (centrada en els tests)

Un cop hem escollit on desenvoluparem les preguntes, s'ha decidit incorporar un servidor de mapes, en concret el que ens proporciona GoogleMaps.

Un primer objectiu d'aquest projecte és descobrir les diverses formes d'avaluar coneixements en un mapa, és a dir, pensar des de el punt de vista del docent o de l'autodidacta quines necessitats i quines funcionalitats vol alhora d'avaluar aquests coneixements amb la informació que dona un servei com GoogleMaps. En conseqüència, volem aprofitar totes les possibilitats pedagògiques que ens pot aportar afegir als qüestionaris mapes interactius (fer *zoom*, assenyalar posicions, desplaçar el mapa lliurement, etc.).

Per a aquesta tasca caldrà fer una anàlisi exhaustiva de requeriments, a nivell de tipus de preguntes, interaccions i mètodes d'avaluació amb la informació proporcionada per un mapa (clàssic o interactiu).

D'aquesta manera partim de la base que al incorporar el servei GoogleMaps, hi haurà una imatge mostrada amb un conjunt de respostes possibles, però a més les possibles respostes podran estar directament sobre el propi mapa.

En segon lloc un cop analitzats els tipus de preguntes que hi ha, serà l'hora de l'acoblament i el treball conjunt entre les tres tecnologies (APIS, GoogleMaps i QTI).

Per a aquest acoblament s'implementarà un middleware que comunicarà APIS amb GoogleMaps en ambdós sentits. Aquest middleware haurà de ser cridat des d'APIS (quan aquest detecti que una pregunta necessita un mapa) per que llegeixi les característiques del mapa indicat, generant el codi necessari per a poder dibuixar-lo en una plana web i crear les interaccions necessàries. Amb el disseny d'aquesta estructura aconseguim no modificar en gran mesura l'estructura i disseny d'APIS.

Un cop APIS i GoogleMaps funcionin de manera coordinada el segon objectiu serà crear tipus de preguntes a sobre d'aquest mapa amb diverses interaccions útils pedagògicament parlant. Aquestes interaccions hauran de ser compatibles amb l'especificació QTI; però aquesta restricció no ens ha de limitar el disseny i implementació d'altres tipus d'interaccions no contemplades per l'especificació IMS QTI.

### **1.3 Planificació del Projecte**

El projecte ha estat conseqüència de la següent planificació dividida en els passos desenvolupats a continuació:

#### *1.3.1 Documentació sobre el projecte proposat*

Comença en el moment que es tanca l'assignació del projecte i es concreta de manera general quins objectius i propòsits es volen assolir a la conclusió.

La durada d'aquesta part del projecte és de gairebé dos mesos fins al final del primer trimestre.

L'objectiu d'aquesta primera fase, es conèixer tot els àmbits que abasta el projecte, totes les tecnologies que s'hauran d'utilitzar i la lectura d'altres projectes emmarcats i relacionats en el mateix àmbit en el qual es mou aquest projecte.

En aquest punt incloem més detalladament els següents passos a destacar:

- a) Conèixer que s'entén per e-Learning, en que consisteix i quines empreses i quins projectes s'estan desenvolupant actualment. Aquest punt té l'objectiu de contextualitzar de manera més general el projecte.
- b) Conèixer els projectes que s'estan desenvolupant actualment, concretament a la Universitat Pompeu Fabra i els projectes de final de carrera de temàtica relacionada d'altres alumnes.

- c) Estudiar els tipus de preguntes que es poden fer, tant test com d'altres tipus. S'ha de veure com funcionen els qüestionaris en aplicacions com, per exemple, Moodle.
- d) Què és i veure com funciona l'API de GoogleMaps i quina informació ens proporciona. S'ha de veure quines eines es poden aprofitar i quines funcionalitats descartarem en un principi per tal de que puguin ser aplicables en l'àmbit educatiu.

### *1.3.2 Coneixement i estudi de les eines que s'utilitzaran*

En aquesta fase del projecte que l'enquadrem des de l'acabament del primer trimestre i que s'estén aproximadament dos mesos, es realitzen les següents tasques amb la finalitat de començar a comprendre, tenir un bon ús i treure el màxim profit de les tecnologies necessàries per al desenvolupament del projecte:

- a) Estudi de l'especificació IMS QTI, així coneixem quines interaccions i quina estructura segueix.
- b) Estudi i comprensió de l'estructura d'APIS. Dins d'aquest punt podem incloure les següents tasques:
  - i. Organització del motor.
  - ii. Tipus de preguntes implementades.
  - iii. Tipus d'interaccions suportades.
  - iv. Determinar la situació actual del projecte APIS per a tal de veure que pot i no pot fer.
- c) Estudi en profunditat de l'API de GoogleMaps. D'aquest procés en podem destacar els següents aspectes.
  - i. Aprendre l'organització de l'estructura.
  - ii. Quines dades es necessiten per a mostrar un mapa.
  - iii. Veure com es situa un mapa dins d'una plana web mitjançant l'API i el codi JavaScript.
- d) Aprofundir els coneixements de JavaScript.

### *1.3.3 Determinar els objectius*

En aquesta etapa es decideix conjuntament amb el tutor del projecte els objectius que cal assolir. La metodologia de treball consisteix en assolir petits objectius abans de començar d'altres més complexes. Així es van afegint de manera gradual més funcionalitats a l'aplicació.

La durada d'aquesta tasca té lloc durant una setmana, un cop s'està familiaritzat amb totes les tecnologies del projecte i sabent les limitacions que poden haver-hi. Alhora els objectius aniran modificant-se a mida que avança el projecte.

### *1.3.4 Anàlisi de requeriments*

S'ha de fer una anàlisi de requeriments d'usuari, funcionals i també els requeriments no funcionals. L'anàlisi de requeriments és un dels punts on s'ha de tenir especial cura, ja que una correcta delimitació i definició dels requeriments farà que els objectius estiguin més clars i les condicions es puguin complir de forma òptima.

Dins dels requeriments d'usuari s'ha d'analitzar quines necessitats tenen tant els docents i avaluadors com els estudiants, per tant, cal fer una anàlisi pedagògica i resoldre les següents preguntes:

- Quins conceptes es volen avaluar sobre un mapa?
- Quins tipus de preguntes són les adients per a una correcta avaluació?
- Quines interaccions són útils des de el punt de vista pedagògic per avaluar geografia sobre un mapa?

Aquesta tasca tindrà una durada d'aproximadament 10 dies i s'ha d'acordar i validar amb el tutor del projecte.

### *1.3.5 Disseny de la solució i implementació d'aquesta.*

En aquest procés cal començar a pensar com haurà de ser la solució i implementar-la. En aquest procés s'han de seguir els següents passos:

- a) Generació d'un middleware que genera codi per a mostrar mapes en una plana web segons unes característiques específiques.
- b) Generació d'una estructura de classes on es representen les dades necessàries en un mapa i per tant la descripció en XML d'aquestes dades.
- c) Integració del Middleware en APIS i realització de la connexió GoogleMaps<--> Middleware<--> APIS

Aquesta fase té una durada d'aproximadament 45 dies (mes i mig) i s'assoleix quan es pot mostrar un mapa de GoogleMaps dins d'un ítem amb el seu corresponent enunciat i possibles respostes (encara que no siguin a sobre del mapa).

### *1.3.6 Generació de diferents tipus de preguntes i interaccions*

Un cop s'ha analitzat quins tipus de preguntes i quines interaccions poden ser les més intuïtives i que afavoreixin l'aprenentatge des de el punt de vista pedagògic arriba el moment d'implementar-les.

En una primera fase s'implementen les interaccions suportades per APIS, com per exemple:

- a) Choice
- b) Multichoice
- c) Order
- d) Match
- e) ...

Per altra banda també s'han de generar interaccions fora de l'àmbit de l'especificació IMS QTI com per exemple.

- a) Interacció amb línies (per a representar rius)
- b) Polígons (com a zones clickables correctes)
- c) Senyalització de rutes,

- d) Emmarcar edificis, localitzacions.
- e) Definir fronteres.

Aquesta tasca ens ocuparà els últims dos mesos de projecte.

### 1.3.7 Redactat de la documentació del projecte

Encara que el redactat de la documentació del projecte s'ha elaborat durant tot el procés de desenvolupament. Les últimes setmanes es dediquen al redactat acurat de la memòria.

## 1.4 Estructura de la memòria

La memòria està estructurada segons els objectius que s'anaven assolint d'acord amb la planificació del projecte de manera incremental.

En primer lloc es fa una petita introducció del context en el qual es mourà el nostre projecte i els objectius que es volen assolir a la conclusió d'aquest. Es donen unes primeres pinzellades a tots els conceptes que després es tractaran amb més profunditat als capítols següents.

Els cinc següents capítols expliquen l'estat de l'art actualment. Concretament els quatre primers ens parlen de tot l'entorn educatiu, primerament, de manera general, que és l'e-Learning i perquè sorgeix la necessitat de crear un camí comú mitjançant els estàndards; per altra banda s'explica l'especificació IMS Question and Test Interoperability amb la qual modelarem els nostres qüestionaris.

Al penúltim capítol d'aquest bloc, s'explica com està estructurat i el funcionament d'APIS (l'eina que permet executar els qüestionaris, interactuar amb ells, validar-los i donar puntuacions segons la resposta donada).

Al capítol sis es descriu l'altra eina utilitzada: GoogleMaps. D'aquest servei s'explica com funciona, quines característiques té, com s'implementa l'API, quines seran les eines que utilitzarem per integrar-lo i poder construir preguntes sobre els mapes, etc.

Un cop s'han explicat les eines emprades per a realitzar el projecte, els següents capítols ens expliquen el procés d'anàlisi, disseny i implementació d'aquest, és a dir, primer es fa una anàlisi de requeriments (funcionals, no funcionals i de sistema). En segon lloc s'explica el disseny i implementació del middleware que connecta APIS amb GoogleMaps i per últim s'explica com es representen les dades d'un mapa de GoogleMaps en un fitxer XML que pugui ser reconegut pel middleware i generar el seu codi corresponent.

Un cop feta la connexió, s'expliquen les interaccions proposades. En aquest capítol s'explica com funcionen i es mostren alguns exemples per a fer-se una idea més clara i concisa del resultat.

Per a concloure el projecte hi ha un capítol dedicat a les conclusions. En aquest apartat s'extreuen les reflexions i conclusions després d'haver realitzat el projecte. D'altra banda es proposen possibles tasques que no han quedat tancades i futurs projectes que es complementarien amb aquest.

## **ESTAT DE L'ART**





## 2. L'e-Learning

---

L' *e-Learning* neix amb la idea d'aprofitar els avantatges que pot donar la xarxa d'Internet en l'àmbit de l'aprenentatge. Per tant, intenta adaptar aquests avantatges a dos conceptes i demandes que abans de l'aparició de l'ús generalitzat d'Internet ja hi havia:

- El procés d'aprenentatge s'ha d'adaptar al ritme i les disponibilitats d'aprenentatge de l'alumne.
- Les eines d'aprenentatge han d'estar disponibles en qualsevol moment que l'alumne les necessiti o les vulgui consultar i independentment de l'espai en el qual estigui en aquell moment.

D'aquests dos conceptes [2] s'extreu que durant molts anys l'alumnat, sobretot d'estudis avançats o complementaris, vol que el seu aprenentatge s'adapti a les seves capacitats i disponibilitats d'estudiar per així adaptar-se al seu propi ritme. En definitiva que pugui estudiar fora dels horaris i terminis establerts i que no hagi d'anar a un lloc en concret i a unes hores concretes per a poder aprendre.

Donades aquestes premisses, podem extreure que aquest model està contraposat d'alguna manera amb el mètode tradicional d'aprenentatge on requereix que aquest es faci en un lloc físic (normalment aules), amb classes presencials i sense cap tipus de disponibilitat per a la mobilitat geogràfica.

L'e-Learning no només contempla l'exposició de material didàctic, també vol ser un espai d'interacció dels diferents alumnes (també entre ells) i els docents.

Les eines que componen aquesta estratègia són:

- Utilitats per a la presentació de continguts:
  - o Textos
  - o Gràfics, imatges i diagrames  
(aquí és on nosaltres intentarem acoblar els mapes de GoogleMaps).
  - o Animacions i vídeos
- Eines de comunicació\* entre la comunitat.
  - o Correu electrònic
  - o Xats
  - o Fòrums
  - o Enquestes
  - o Feedbacks

Els principals avantatges de l'*e-Learning* són els següents [1][3][4][5]:

- Escalable: el desenvolupament ha de créixer de forma modular tant en mida com en les funcionalitats que es generin per tal de poder-se adaptar a les necessitats futures.
- Independència: Ha de poder executar-se sota diferents softwares, hardwares i SO sense cap tipus de modificació en la seva implementació.
- Estàndards oberts: La plataforma s'adhereix a aquests estàndards.

\*Cal especificar com a comunicació entre comunitat, la comunicació entre els diferents alumnes i els alumnes amb els avaluadors, professors i tutors.

- Flexible: Que sigui una arquitectura oberta, fa que el disseny pugui adaptar-se a les necessitats de cada organització, adaptant-lo i reorganitzant-lo a la seva mida
- Fàcil de mantenir: Al ser un desenvolupament modular, el seu manteniment es pot fer de manera transparent.
- Integrat: Es pot integrar amb pràcticament qualsevol aplicació, ja que està desenvolupat amb una arquitectura oberta com Java. Només cal afegir els connectors necessaris com XML.

La plataforma abasta tot el procés de formació, des de la creació de continguts didàctics i cursos fins el seguiment personalitzat de cada usuari.

L'e-Learning, dóna un espai comú a una comunitat molt important de persones [6], tant usuaris (docents i alumnes) com desenvolupadors. Degut a aquest gran volum de persones interessades en comunicar-se i participar d'un projecte comú cal trobar una manera de que totes les parts es puguin entendre per a arribar a aquest objectius consensuats.

### 3. Adaptació als estàndards

---

Tal com hem esmentat abans, volem incorporar un servei gratuït com GoogleMaps a un estàndard ja fet que no el contempla, per tant haurem d'adaptar-nos a aquest estàndard i intentar fer que tot sigui compatible amb l'especificació ja feta. Per això primer de tot cal saber la importància de tenir i adoptar un estàndard en l'àmbit educatiu i en altres àrees del software.

#### 3.1 Motivació per a generar estàndards

Des de que existeix la comunicació humana, i en conseqüència la diversitat de maneres d'expressar-se, esdevé de capital importància trobar un espai comú per a poder entendre'ns.

Podem posar diversos exemples com el mateix llenguatge natural. Qualsevol idioma necessita un estàndard que sigui útil per a tota la comunitat. Totes aquelles persones que utilitzen aquesta llengua s'han de poder entendre amb la resta de persones que comparteixen la seva mateixa manera de parlar.

El món informàtic no és aliè a aquest fenomen, i adquireix especial importància en el moment que es poden utilitzar diferents llenguatges de programació o diferents estils d'arquitectura per a desenvolupar la mateixa funcionalitat.

Per tant qualsevol canvi o millora que es vulgui implantar i ser acceptat per la resta de la comunitat ha d'adaptar-se a unes regles bàsiques decidides en comú acord.

D'aquesta manera les diferents empreses, entitats o persones podran desenvolupar un mateix producte sense provocar que tota la resta de la comunitat hagi de tornar-se a adaptar.

A aquesta sèrie de regles decidides en comú acord se'ls anomena estàndards. Adoptar-ne un és garantia d'una sèrie d'avantatges [9]:

- *Reutilització*: Fomentar el desenvolupament de continguts educatius integrals i d'elevada qualitat que siguin intercanviables i flexibles per adaptar-se a diversos ambients d'entorns d'aprenentatge diferents al inicialment dissenyat.
- *Intercanviabilitat*: Fer possible la reutilització, s'ha de poder ajuntar i separar amb facilitat i rapidesa.
- *Interoperabilitat*: Garantir la qualitat de que hi hagi compatibilitat tecnològica de continguts i serveis entre les diferents plataformes on s'executa i els entorns d'e-Learning.
- *Durabilitat*: Assegurar que els recursos siguin accessibles encara que els sistemes i plataformes canviïn en un futur. També cal garantir que els usuaris i compradors no es trobaran atrapats per les prestacions de tecnologies sota llicència propietària.

Les eines que s'han de crear per assolir aquests objectius són:

- *Editors*: Són eines per a crear documents que segueixen les normes dictades per l'estàndard, facilita la creació dels documents, sense que

l'usuari hagi de conèixer en profunditat l'especificació. Per tant els editors donen una interfície més amigable i entenedora alhora de generar els documents. El propi editor és qui s'ocupa d'adaptar a l'estàndard els elements concretats per l'usuari.

- Reproductors o *Players*: Són eines amb la capacitat d'interpretar l'estàndard i mostrar-lo. També fa que les interaccions entre els diferents usuaris siguin possibles.
- Motor: Posa en marxa l'aplicació, duu a terme i encapsula les funcionalitats perquè es puguin fer ús des d'altres programes

Un cop hem vist la motivació, tots aquests principis també són aplicables a qualsevol àmbit informàtic, tant software com hardware, i així com a l'entorn e-Learning. A més, en el moment actual, amb la gran acceptació del fenomen d'Internet com a eix comunicatiu i vertebrador de la societat (gràcies als estàndards), és de capital importància l'estandardització ja que el volum de persones i comunitats que s'han d'entendre creix exponencialment.

Durant el projecte ens anirem referint a estàndards i especificacions, per tant cal diferenciar aquests dos conceptes i els punts en comú que tenen [10]:

- Especificació: Són un conjunt de declaracions detallades de forma unívoca que es desprenen dels requeriments funcionals d'allò que es vol construir, instal·lar o manufacturar.
- Estàndard: És un patró, una tipificació o una norma per a definir quelcom i poder realitzar-lo. Fins aquí la definició és semblant a la d'una especificació, la diferència roman en que estan produïts i acceptats per organismes internacionals reconeguts per un o diversos governs nacionals.

### 3.2 El procés d'estandardització

Per arribar a l'acceptació d'una determinada proposta (estructura de desenvolupament) cal desenvolupar una sèrie de passos per a aconseguir que s'aprovi [11] [28]:

- Fase de recerca i investigació: L'estàndard ha de sorgir d'una necessitat de mercat en una determinada àrea. Donada aquesta demanda cal estudiar, desenvolupar i encetar propostes per a solucionar-lo. Es fa una anàlisi de requeriments, casos d'ús, demandes específiques, etc.
- Selecció i documentació: Analitzades les propostes, cal escollir una que satisfaci les necessitats el més àmpliament possible a la comunitat i fer una documentació exhaustiva d'aquesta solució complimentada amb una justificació d'aquesta envers les altres. ARIADNE o IMS són organitzacions que s'encarreguen d'aquesta tasca. Aquesta selecció i documentació cal fer-la amb un consens d'àmplia majoria.
- Proves i test: L'especificació es posa a prova en diferents situacions i àmbits analitzant els punts forts i dèbils d'aquesta. Cal complementar-lo analitzant el nivell d'acceptació per part de la comunitat en aquests diferents entorns.

- Revisió: A mida que es fa la prova i *testing* de manera iterativa es van fent modificacions, millores i adaptacions per tal de superar amb garanties la fase de validació.
- Validació i acreditació per part d'organismes internacionals: En aquesta fase es revisa per part d'organismes com IEEE, ISS, ISO o CEN. Un cop revisat es valida i es dona l'acreditació.

*NOTA:* És àmpliament acceptat per tota la comunitat l'afirmació de que encara no hi ha estàndards d'e-Learning, només hi ha una sèrie de grups desenvolupant especificacions. A la pràctica però, no es fan diferències entre un concepte i l'altre. D'aquesta manera, sovint es troba que les especificacions d'e-Learning són anomenades estàndards (el qual no és del tot erroni ja que algunes d'aquestes especificacions han estat adoptades per importants grups de desenvolupament i, per tant, poden considerar-se com estàndards *de facto*).

### 3.3 Els estàndards a l'e-Learning

A l'àrea educativa, les especificacions són formalment enviades al LTSC (Learning Technology Standards Committe), que és un comitè que està especialitzat en e-Learning de l'IEEE, única organització acreditada d'estandardització.

La problemàtica existent era que cadascuna de les empreses i persones que generaven alguna funcionalitat o projecte en aquest camp estava feta sota criteris totalment disjunts, en conseqüència totes les millores esdevenien incompatibles les unes amb les altres i no es podien transferir entre diferents sistemes d'ensenyament.

D'una manera més simple, si algú creava un curs, sota un format definit en XML i el reproduïa d'una determinada manera, aquest projecte no podia ser aprofitat per a qualsevol membre de la resta de la comunitat, ja que no disposava de cap manera de reproduir-lo a la seva màquina si no era sota les premisses de l'altre desenvolupador.

Sota aquestes condicions era impossible que aquests projectes tinguessin èxit (acceptació) i fossin d'ús generalitzat en el món de l'aprenentatge virtual. Tant és així, que era necessari per a la subsistència d'aquests projectes definir uns estàndards comuns i un compromís per tal de que tots desenvolupessin sota aquests criteris preestablerts.

D'aquesta manera es començaren a crear estàndards i especificacions per a tal de solucionar el problema, aquests estàndards i especificacions es dissenyaven amb l'objectiu de facilitar la descripció dels projectes, empaquetar-los, seqüenciar-los i facilitar el lliurament del material.

Més concretament en l'àmbit de l'educació serien estandarditzats els continguts educatius, les activitats d'aprenentatge i la informació de l'alumnat mitjançant els repositoris, fent així més eficients els processos de producció i reduint sensiblement els esforços de desenvolupament.

Cal destacar també que tot el moviment d'e-Learning i concretament els estàndards internacionals resultants són oberts i per tant totes les seves especificacions són públiques.

### 3.4 IMS Global Learning Consortium

[4] És un Consorci conformat per membres d'organitzacions educatives on també hi participen empreses públiques i privades.

El seu objectiu bàsic és promoure i desenvolupar especificacions obertes per a facilitar les activitats d'aprenentatge on-line.

IMS Global Learning Consortium pren com a punt de partida el treball fet per IEEE LTSC (Learning Technology Standards Committee). El seu treball es caracteritza pel disseny d'un format que pogués posar en pràctica les recomanacions de la IEEE i la AICC.

Els seus primers treballs del IMS es varen centrar en la definició d'un model i una arquitectura per a l'aprenentatge distribuït. Més endavant els seus esforços es van reorientar ràpidament al notar que primer necessitaven un model de dades adequat per a poder descriure els recursos, les estructures i altres elements manegats per components, definicions de test i qüestionaris, especificacions de perfils d'alumnes i gestió de grups, recomanacions per objectes educatius distribuïts, organització de cursos sota enfocaments pedagògics i seqüenciamnt.

Per tal d'arribar a aquest objectiu, es va definir un tipus de fitxer XML per a la descripció dels continguts dels cursos i així qualsevol LMS podia carregar el curs, llegint el seu fitxer de configuració IMSMANIFEST.XML.

Una de les seves principals iniciatives i que serà la que nosaltres utilitzarem per a desenvolupar aquest projecte és la Interoperabilitat de Preguntes i Proves (Question and Test Interoperability, QTI) que serà on nosaltres incorporarem els mapes per tal de poder ser avaluats.

Al següent capítol explicarem com funciona aquesta especificació proposant una estructura de dades XML per a codificar preguntes i test *online*. L'objectiu d'aquesta especificació és fer possible l'intercanvi d'aquestes proves i dades d'avaluació entre diferents LMS.

## 4. IMS Question & Test Interoperability

---

### 4.1 Motivació

La definició més formal per a l'especificació IMS QTI és la descripció d'una estructura bàsica per a la representació de preguntes (*items*) i qüestionaris (*assessments*) i els seus corresponents resultats.

Està basat en XML, ja que al ser un llenguatge estàndard, de gran abast i molt flexible pot ser utilitzat per a codificar models de dades d'aplicacions basades en Internet i sistemes distribuïts. També es va decidir d'utilitzar-lo ja que així podria ser adoptat el més àmpliament possible per la comunitat.

L'especificació suporta preguntes de text, tests (simples i complexos), i preguntes amb imatges sempre i quan estiguin definits sense ambigüitats.

En qualsevol procés d'aprenentatge els qüestionaris estan àmpliament acceptats per la nostra societat com una eina per avaluar si un alumne ha adquirit els conceptes i coneixements que s'han impartit durant el curs. També, com hem dit a la introducció, esdevé una eina d'autoavaluació per al propi alumne, d'aquesta manera la persona pot valorar com ha assimilat el temari i pot reforçar les parts de la matèria que consideri que no domina amb suficiència.

Amb l'aparició de les eines informàtiques es dona la possibilitat de que aquests tipus de preguntes puguin ser corregides d'una manera automàtica mitjançant les eines adients. D'aquesta manera un professor pot generar un grup de preguntes que el sistema informàtic pot utilitzar per a preparar qüestionaris i corregir-se automàticament.

Per això l'objectiu d'aquesta especificació és donar una eina i una solució a la creació, gestió i realització de qüestionaris on-line. Amb la realització i acceptació d'una especificació com aquesta es pot resoldre el problema que pot comportar l'esforç necessari per a realitzar, per part de l'instructor, les preguntes i els inconvenients provocats per un possible canvi de plataforma.

Seguint amb els objectius dels estàndards, l'IMS QTI vol aconseguir que quan hi ha un esforç en la creació de qüestionaris hi hagi la possibilitat i la recompensa de compartir aquest esforç donant la possibilitat d'intercanviar-los o poder crear repositoris de preguntes.

Aquestes preguntes també haurien d'estar classificades per matèries i grau de dificultat per a simplificar la seva localització i reutilització en la conformació de nous qüestionaris.

El format de IMS QTI està basat en XML, per tant fa ús d'estàndards àmpliament utilitzats en l'àmbit empresarial i tècnic, donant la possibilitat d'utilitzar les mateixes preguntes entre diversos sistemes de gestió d'aprenentatge (LMS), entre diversos sistemes d'avaluació electrònica independents i la integració en un únic LMS de preguntes i qüestionaris desenvolupats amb diverses eines. De la mateixa manera es proposa un sistema coherent per a que els usuaris estiguin informats de quin és el resultat d'una avaluació.

La construcció de repositoris en IMS QTI per a preguntes que puguin ser intercanviables i reutilitzables per a diversos sistemes LMS, la generalització

d'aquest tipus de magatzems i la seva lliure disposició en format compatible amb altres plataformes pot simplificar molt la tasca de docents [7].

## 4.2 Evolució de l'especificació i situació actual

IMS Question and Test Interoperability és una de les primeres especificacions que es van començar a desenvolupar. De fet, els primers treballs tenen com a data d'inici l'any 1999. A més, cal dir que el potencial d'aquesta especificació encara no ha arribat al seu punt més àlgid degut a diversos factors com la seva complexitat i les poques eines informàtiques que poguessin posar en pràctica l'especificació. Habitualment les eines existents només posaven en pràctica un petit conjunt de totes les possibilitats que ofereix l'especificació.

IMS QTI ha llençat tres versions que han obtingut gran èxit i acceptació per part de la comunitat desenvolupadora [28]:

- **IMS QTI versió 1.2:** Fa la seva aparició a l'any 2002 i podríem dir que és la última versió finalitzada completament. Aquesta versió dona la possibilitat de generar preguntes individuals i qüestionaris complets. Un cop es va publicar l'especificació, van sorgir diversos problemes a l'hora d'implementar-la. Encara es va publicar un suplement, que es va anomenar IMS QTI v1.2.1, però els problemes que es van detectar necessitaven de grans canvis i que aquestes modificacions provocarien que es perdés gran part de la compatibilitat amb versions anteriors. A més, algunes altres parts necessitaven una revisió o una expansió.
- **IMS QTI versió 2.0:** Aquesta especificació neix al 2005 amb la voluntat de ser compatible amb tots els projectes que s'estaven desenvolupant en paral·lel en aquell moment. D'aquesta manera es van començar a solucionar els problemes que s'arrossegaven de l'anterior versió. Per a simplificar el procés d'adopció i permetre un treball raonable, aquesta especificació va concentrar-se només en les preguntes individuals i no actualitzà aquelles parts que estaven relacionades amb la composició de les preguntes, és a dir, amb la creació de qüestionaris complets. En aquesta versió es comença a deixar de banda la presentació final de les preguntes i es centra en definir tipus d'interaccions amb l'usuari. A més de formalitzar totes aquestes interaccions, introdueix un tipus d'interacció propi per a poder estendre el model i crear noves formes d'interactuar amb nous tipus de qüestions. Per tant, QTI no és un model complet i tancat, si no que dona la possibilitat de que si una entitat considera oportú estendre'l o ampliar-lo per adaptar-lo a les seves necessitats pugin fer-ho encara que no siguin reconegudes en un altre sistema QTI.

Aquesta nova versió també incorpora plantilles de qüestions per a crear famílies de preguntes similars però en les que hi ha parts que poden ser variables.

També introdueix preguntes adaptatives, és a dir, donen la possibilitat de determinar el nombre d'intents per a resoldre la qüestió formulada. Una aplicació d'aquestes possibilitats són per exemple que l'alumne pugui canviar la seva resposta en funció de la retroacció donada per l'avaluador mitjançant pistes que puguin modificar la seva resposta.



- **IMS QTI versió 2.1:** Aquesta versió sorgeix a l'any 2006 i està actualment en procés d'evolució en mode esborrany sobre el que la comunitat, tant educativa com tècnica, pugui opinar, discrepar o millorar.

El principal objectiu d'aquesta nova versió és seguir amb el procés de simplificació i evolució de l'especificació, aquesta vegada donant suport als qüestionaris complets i l'intercanvi dels resultats dels mateixos. Així també s'inclou informació per a clarificar la compatibilitat i l'ús d'IMS QTI amb algunes altres especificacions ja existents.

### 4.3 Característiques tècniques de l'especificació

Com hem dit abans l'estàndard s'ha conceptualitzat de manera totalment independent en tres vessants:

**Les preguntes (*items*):** Les preguntes individuals poden ser utilitzades com a un recurs educatiu independent, per exemple, com un recurs més dins d'un paquet IMS.

**Els qüestionaris (*assessments*):** Els qüestionaris són agrupacions de preguntes que fan possible fer un resum de les avaluacions aconseguides a les qüestions (*items*) individuals en una única avaluació del qüestionari, és a dir, aquests estan conformats per un número indeterminat d'*items*.

**L'avaluació dels qüestionaris:** La interacció dels alumnes amb les preguntes individuals i els qüestionaris generaran diferents registres d'informació que poden ser emmagatzemats per a estudis posteriors.

Com que l'objectiu d'aquest projecte és generar qüestions individuals sota una característica comuna (els mapes), ens interessa especialment veure més en profunditat com estan fets els *items* i així veure en més detall la manera de poder incorporar el servei com un element més dins de l'especificació.

### 4.4 Les preguntes

Són les unitats mínimes que poden ser intercanviables utilitzant QTI, per tant, els *items* no poden ser conformats per altres *items*.

Aquestes preguntes individuals són auto-contingudes, és a dir, dins d'elles mateixes inclouen tota la informació necessària per a la seva presentació a l'alumne i la seva correcció automàtica. Tota la informació relativa a la presentació ha estat agrupada en el cos de la pregunta. A la presentació de cada pregunta estan relacionats els següents aspectes:

**L'enunciat de la pregunta:** Òbviament, cada qüestió ha de contenir el seu respectiu enunciat i de manera addicional pot contenir material explicatiu complementari que faci possible al docent indicar el context en el qual es realitza la pregunta (imatges, gràfics, text addicional... i al finalitzar aquest projecte els mapes).

En la versió 2.X de l'especificació, els continguts que podem utilitzar dins d'aquest grup, segueixen l'estàndard XHMTL, és a dir, contingut web, i a més, fa possible utilitzar l'estàndard MathML per a la representació d'equacions matemàtiques.

Com hem dit abans és al material contextualitzador de l'enunciat on el nostre projecte pretén inserir els mapes de GoogleMaps.

**La construcció de la resposta:** De manera addicional a l'enunciat de la pregunta, ha d'haver-hi de cara a l'alumne l'equivalent del llapis i el paper per a construir la resposta. En el cas d'IMS QTI v2.X s'ha introduït el concepte d'interacció.

Aquí podem veure l'estructura dels *Items*:

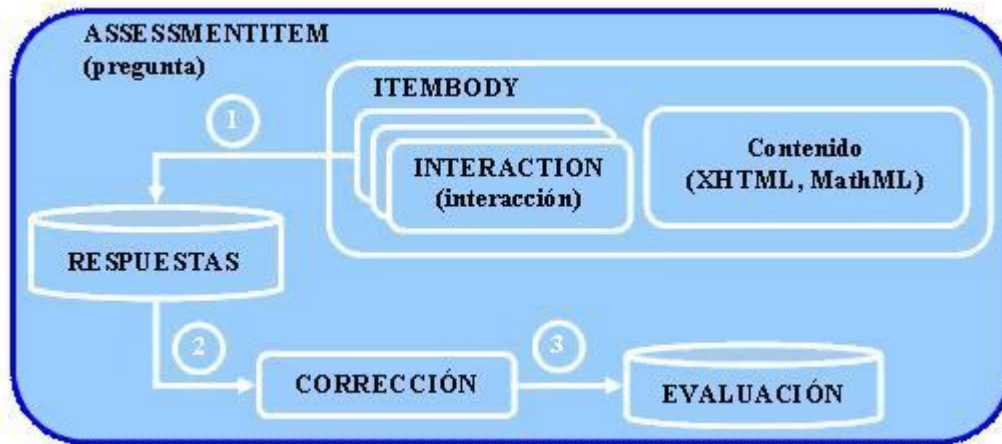


Fig 1: Estructura d'un assessmentItem i el seu procés d'avaluació

Adicionalment a la definició de la presentació de la pregunta i juntament amb la definició de la interacció es podrà recollir la resposta, és necessari especificar com corregir la pregunta. El funcionament de la correcció és simple i directe. Quan un alumne ha de respondre, l'eina presentarà la pregunta i la interacció a l'alumne. Com a resultat d'aquesta interacció obtindrem una representació de la resposta de l'alumne. Aquestes respostes es faran servir com a informació d'entrada per al procés de correcció, generant finalment l'avaluació.

## 4.5 El nucli

Podem dir d'un mode inicial que aquesta especificació s'estructura en un nucli que està conformat i que nosaltres explicarem a partir d'aquest gràfic esquemàtic:

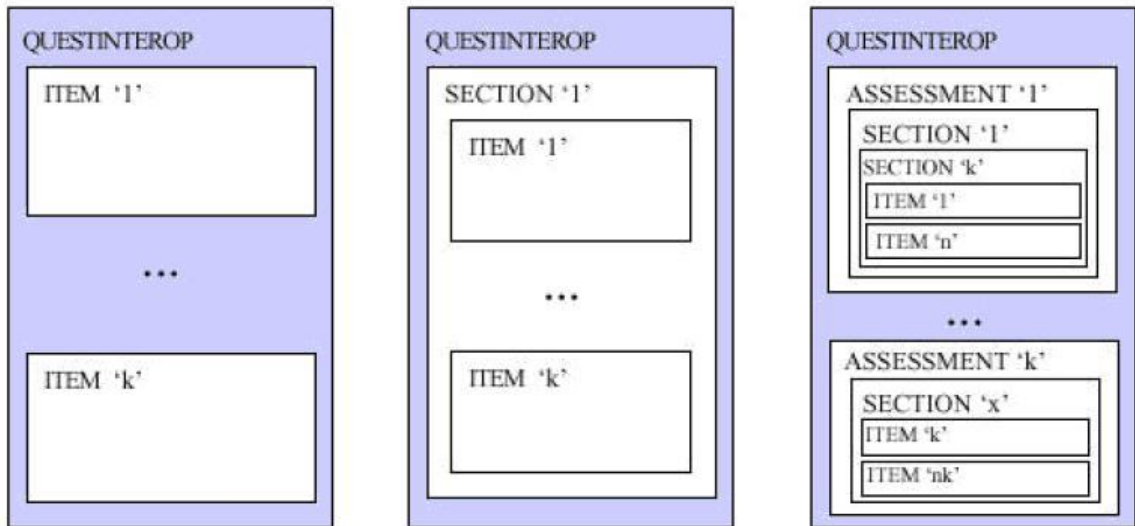


Figura 2 Esquema de l'estructura de l'especificació IMS QTI

Com ja hem especificat com està format un ítem, també caldrà definir les altres parts de l'especificació:

*Sections:* Un arxiu XML QTI pot contenir diverses seccions on cadascuna alhora pot contenir diversos *ítems* o seccions.

*Assessments o test:* Només pot haver-hi un assessmentTest per instància representant el qüestionari. Aquest està dividit en una o més parts que alhora poden estar dividides en seccions i sub-seccions. L'única restricció és que cada Assessment ha de tenir com a mínim una secció

## 4.6 Les interaccions

Aquest punt és de rellevant importància, ja que cal saber quins tipus d'interaccions es defineixen i es conceptualitzen a l'especificació IMS QTI per tal de poder veure després els avantatges que tindrà incorporar un servei com GoogleMaps respecte aquestes interaccions i si d'alguna manera es poden complementar.

Abans de tot però, començarem per una descripció detallada de les interaccions.

Formalment, a IMS QTI no hi ha representat el concepte de "tipus de pregunta", en comptes d'això, es formalitza el concepte *tipus d'interacció*. Les interaccions fan possible al professor especificar les eines que tindrà l'alumne per a poder donar una resposta a la pregunta formulada.

Existeixen tantes interaccions com tipus de preguntes i eines per respondre es puguin imaginar. A continuació descriurem els tipus d'interaccions formalitzades per IMS QTI que poden haver-hi dins d'una pregunta [11].

Abans de tot cal diferenciar dos tipus bàsics d'interacció:


- *Interaccions en línia*: Aquesta interacció es pot incloure dins o enmig de l'enunciat de la pregunta.
- *Interaccions en bloc*: Aquestes interaccions estan dissenyades per a ser presentades de manera independent a l'enunciat de la pregunta.

#### 4.6.1 Interaccions simples

Les interaccions simples són aquelles que es realitzen en base a la selecció d'una opció o diverses opcions disponibles. Les interaccions que estan incloses a aquesta categoria són les següents:

*choiceInteraction*: Dóna la possibilitat a l'alumne d'escollir una o diverses opcions sobre un conjunt múltiple de respostes. Aquesta interacció fa possible que l'ordre de com es mostren les respostes per pantalla compleixin un ordre aleatori per a cada intent de l'alumne.

Look at the text in the picture.



NEVER LEAVE LUGGAGE UNATTENDED

What does it say?

You must stay with your luggage at all times.

Do not let someone else look after your luggage.

Remember your luggage when you leave

Next End Exam

*orderInteraction*: En aquesta interacció l'objectiu de l'alumne és reordenar un conjunt de solucions proporcionada. Permet també fer que totes les solucions no siguin respostes correctes, per tant obliga a l'alumne, escollir un conjunt de respostes bones entre el total de respostes i després reordenar-les.

The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order?

1	Jenson Button
2	Rubens Barrichello
3	Michael Schumacher

submit

*associateInteraction*: Aquesta interacció fa possible crear associacions per parelles entre un conjunt de possibles respostes.

### SHAKESPEARIAN RIVALS

Hidden in this list of characters from famous Shakespeare plays are three pairs of rivals. Can you match each character to his adversary?

Lysander	Prospero
Antonio	
Capulet	Montague
Demetrius	

*matchInteraction*: És molt similar a l'anterior, l'única diferència roman en que l'associació no cal que es faci per parelles.

### CHARACTERS AND PLAYS

Match the following characters to the Shakespeare play they appeared in:	The Tempest	Romeo and Juliet	A Midsummer-Night's Dream
Prospero	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Capulet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Demetrius	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lysander	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*gapMatchInteraction*: Fa possible definir un conjunt d'espais dins d'un enunciat de la pregunta a mostrar de cara a l'alumne. S'associa a cadascun dels espais una de les possibles respostes donades.

**RICHARD III (TAKE 1)**

**Identify the missing words in this famous quotation from Shakespeare's Richard III.**

Now is the  of our discontent  
Made glorious  by this sun of York;  
And all the clouds that lour'd upon our house  
In the deep bosom of the ocean buried.

**Use the table below to select the missing words.**

	winter	spring	summer	autumn
Word 1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Word 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### 4.6.2 Interaccions de text

Aquest tipus d'interaccions s'anomenen així ja que l'alumne només podrà donar com a resposta una paraula, una frase o un paràgraf de text.

En algunes d'aquestes interaccions podria ser que no hi hagués correcció automàtica, ja que cal una revisió i/o lectura per part del professor o avaluador.

Els tipus d'interaccions definides en aquesta categoria són:

*inLineChoiceInteraction*: Aquesta és una interacció en línia, es pot situar un espai incrustat al mig d'un text proporcionant un conjunt de paraules per emplenar aquest espai. Aquestes paraules són definides per l'avaluador.

**RICHARD III (TAKE 2)**

**Identify the missing word in this famous quotation from Shakespeare's Richard III.**

Now is the winter of our discontent  
Made glorious summer by this sun of  ;  
And all the clouds that lour'd upon our house  
In the deep bosom of the ocean buried.

*textEntryInteraction:* Com a la interacció anterior hi ha espais enmig de l'enunciat que cal que l'alumne ompli. En aquests tipus d'interaccions l'alumne és lliure per a posar el text que vulgui, alhora de corregir-lo poden haver correccions automàtiques o manuals, les automàtiques estan pensades per a que el docent introdueixi expressions regulars vàlides com a resposta. També es pot especificar la mida màxima de la resposta.

**RICHARD III (TAKE 3)**

Identify the missing word in this famous quotation from Shakespeare's Richard III.

Now is the winter of our discontent  
Made glorious summer by this sun of York  ;  
And all the clouds that lour'd upon our house  
In the deep bosom of the ocean buried.

*extendedTextInteraction:* En aquesta interacció l'alumne pot escriure una redacció lliure com a resposta. Clarament aquesta resposta ha de ser validada manualment pel professor.

**WRITING A POSTCARD**

Read this postcard from your English pen-friend, Sam.

Here is a postcard of my town. Please send me a postcard from your town. What size is your town? What is the nicest part of your town? Where do you go in the evenings?

Sam.

**Write Sam a postcard. Answer the questions. Write 25-35 words.**



*hottextInteraction:* L'objectiu d'aquesta interacció és que l'alumne seleccioni parts d'un text que estaran ressaltades a l'enunciat de la pregunta.

**IDENTIFYING SENTENCE ERRORS**

**Select the error in the following passage of text (or *No Error* if there is none).**

Sponsors of the Olympic Games  **who bought** advertising time on United States television  **includes**  **at least** a dozen international firms  **whose** names are familiar to American consumers.  **No error.**

#### 4.6.3 Interaccions gràfiques

Com que en aquest projecte treballarem amb un servei que ens retorna fotografies, hem de conèixer quines possibilitats ens dona l'especificació a nivell d'interacció amb imatges, encara que després, les interaccions amb el mapa no siguin contemplades específicament com a interaccions sobre imatges.

*hotspotInteraction:* L'objectiu d'aquesta interacció és donar a l'alumne un conjunt de punts seleccionables (hotspot) sobre una imatge utilitzada com a fons de l'enunciat. L'alumne haurà de seleccionar un o més d'un d'aquests punts per a donar la seva resposta. En el servei GoogleMaps seria equiparable a situar marcadors a sobre el mapa.

**UK AIRPORTS (TAKE 1)**

**The picture illustrates four of the most popular destinations for air travellers arriving in the United Kingdom: London, Manchester, Edinburgh and Glasgow.**

**Which one is Glasgow?**





*selectPointInteraction:* En aquest tipus d'interacció l'alumne selecciona un o diversos punts de la imatge utilitzada com a fons de l'enunciat. Al contrari que a la interacció anterior, no se li donarà a l'alumne cap zona seleccionable o ressaltada.

**WHERE IS EDINBURGH?**

**Mark Edinburgh on this map of the United Kingdom.**

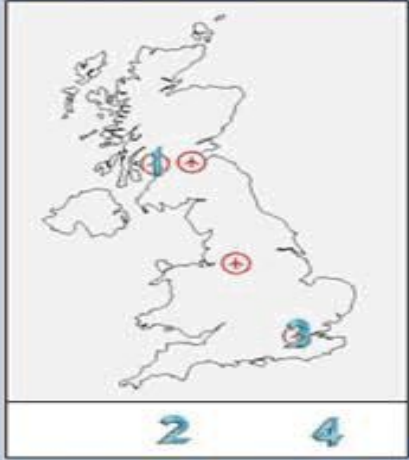


*graphicOrderInteraction:* En aquesta interacció es mostra un conjunt de punts "calents" sobre la imatge. L'objectiu és que l'alumne ordeni aquests punts adequant-los a la resposta.

**FLYING HOME**

**Lorna is flying back home to the UK. Ideally, she would like to fly in directly to her home town of Glasgow. Edinburgh is her second choice and, if necessary, she could fly into London and pick up an internal connecting flight. Although she has been offered a cheap flight to Manchester it remains her least favourite option as connecting flights to Glasgow are not very reliable from there.**

**Mark the airports shown on the map according to Lorna's preferences.**

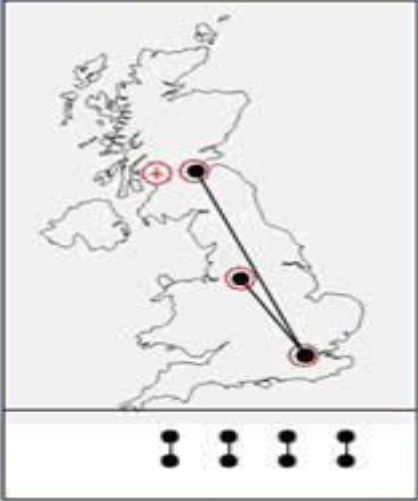


*graphicAssociateInteraction:* Aquesta interacció ens mostra un conjunt de zones seleccionables. L'objectiu de la interacció és permetre a l'alumne la creació de parelles d'associació entre punts seleccionables.

**LOW-COST FLYING**

**Frizz, a new low cost airline, already operates a service connecting Manchester and Edinburgh but has recently opened two new routes: a service between London and Edinburgh and one between London and Manchester.**

**Mark the airline's new routes on the airport map:**



Drag the markers by their ends to connect the appropriate points on the image

*graphicGapMatchInteraction:* L'objectiu d'aquesta interacció és que l'alumne construeixi parells entre els punts seleccionables dins la imatge i també que triï el conjunt de respostes correctes d'entre totes les possibilitats.

**AIRPORT TAGS**

**The International Air Transport Association assigns three-letter codes to identify airports worldwide. For example, London Heathrow has code LHR.**

**Some of the labels on the following diagram are missing: can you identify the correct three-letter codes for the unlabelled airports?**



EBG    GLA    NCH  
CBG    EDI    MAN

*positionObjectInteraction*: En aquesta interacció, l'alumne col·locarà una imatge sobre alguna zona indicada en una altra imatge. És semblant a la d'omplir espais en blanc en mode text, però amb imatges.



#### 4.7 Els qüestionaris

A continuació es mostra el diagrama de casos d'ús en el qual hi ha indicats els possibles actors que poden interactuar amb la composició de preguntes.

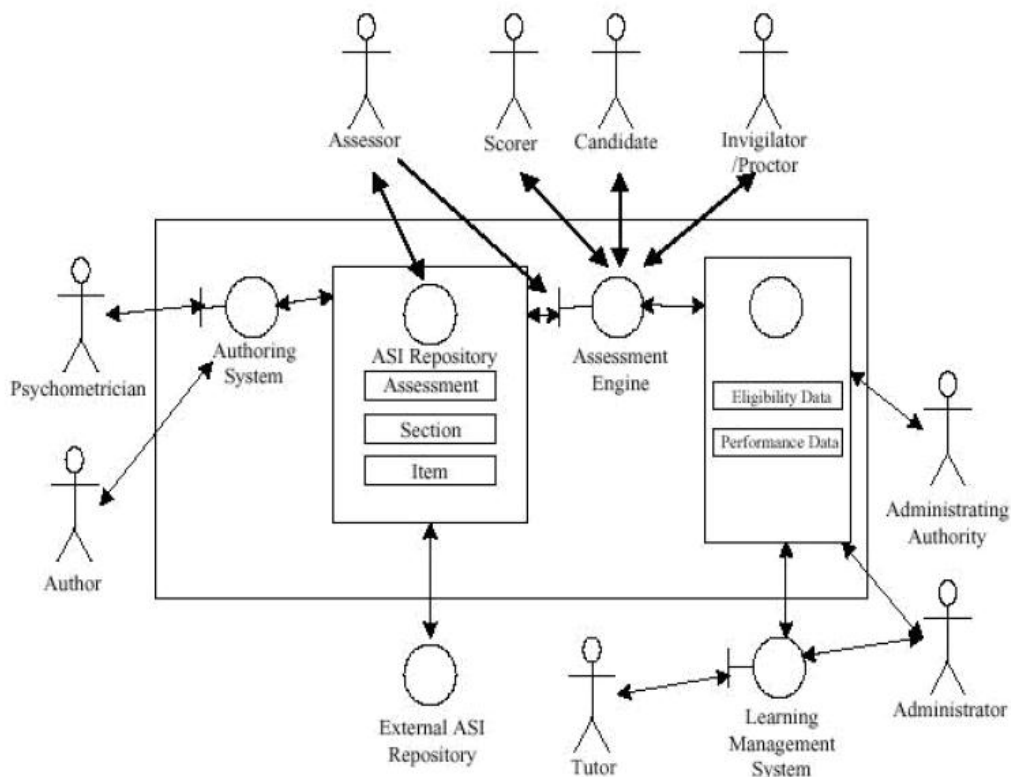


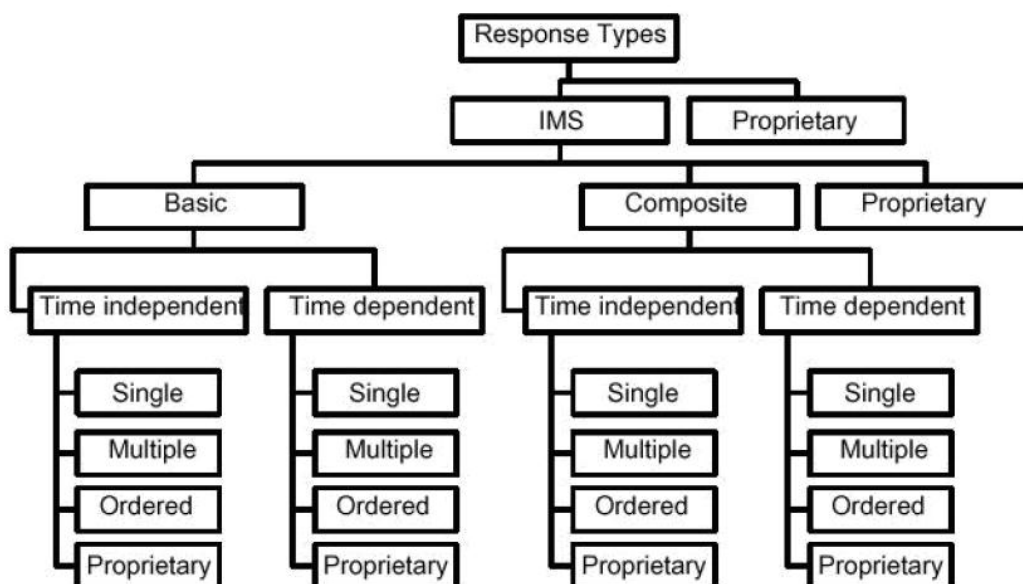
Figura 3: Casos d'ús en un sistema de qüestionaris a IMS QTI

Tal com es veu en aquest esquema de casos d'ús, els principals components del sistema qüestionaris són:

- *Authoring System*: És el procés que suporta la creació i l'edició de qüestionaris, seccions i preguntes (ASIs).
- *Assesment Engine*: És el procés que suporta l'avaluació i la qualificació de les respostes de manera que generi qualificacions i consells que reforcin l'estudi.
- *Learning Management System*: Sistema i procés responsable de la gestió completa de l'arquitectura del curs d'aprenentatge.
- *Candidate and Repository*: Es tracta d'una base de dades amb les especificacions de cara a l'alumne.
- *ASI Repository*: És una base de dades local de qüestionaris, seccions i preguntes.
- *External Asy Repository*: Base de dades externa que ha d'ésser construïda i importada seguint les especificacions IMS QTI.

## 4.8 Taxonomia

Per últim i tancar aquesta secció sobre IMS Question and Test Interoperability, especifiquem, a mode de resum, la taxonomia que té l'especificació:



Com hem enunciat abans a banda de les qüestions preestablertes hi ha també interaccions propietàries, que cada entitat pot definir i modificar per adaptar-lo a les seves necessitats. Aquesta possibilitat fa que escollim aquesta especificació per al nostre projecte, ja que ens dóna certa flexibilitat per a poder incorporar el nostre servei.

Per acabar, cal remarcar que cadascun dels *items* es pot definir amb un temps específic o amb temps il·limitat per respondre . Això pot influir en el mètode d'avaluació.

## 5. Assessment Provision through Interoperable Segments

---

### 5.1 Definició del projecte

El projecte Assessment Provision through Interoperable Segments (APIS) és un dels primers projectes en codi obert que implementa l'especificació IMS Question and Test Interoperability (IMS QTI). Per tant és el primer cop que es posa a la lliure disposició un codi per a construir o avaluar qüestionaris i preguntes; el principal avantatge és que cada cop que es vulgui implementar una nova funcionalitat, no caldrà començar de zero, ja que hi ha a disposició pública aquest projecte *open-source*.

S'emmarca el naixement d'aquest projecte a l'any 2004 a la University of Strathclyde sota el finançament de JISC (el qual porta diversos projectes en paral·lel relacionats amb temes d'e-Learning).

### 5.2 Estructura

El projecte APIS implementa un sistema modular per a la generació de qüestionaris i tests seguint les especificacions IMS QTI. Aquest projecte proveeix de llibreries bàsiques per a construir, lliurar i determinar qüestionaris amb *ítems* que implementa interaccions de resposta.

El motor està desenvolupat en un principi: fer que els mòduls addicionals siguin els que proporcionin les funcionalitats. Per tant APIS fa possible que es puguin incorporar en el sistema base tots els mòduls que es vagin desenvolupant, cadascun d'ells cobrint una part de l'especificació tal com està definit en IMS QTI.

Per altra banda, implementa i proporciona una interfície gràfica de fàcil utilització i molt comprensible per a usuaris no iniciats en informàtica i TIC's, a més de donar un entorn agradable, ordenat i estructurat per als desenvolupadors. Proporciona preguntes amb interaccions intuïtives a l'abast de tothom.

També incorpora una estructura de gestió per administrar quan s'han de mostrar les preguntes, com s'han de mostrar, si s'han de mostrar les correccions o si s'han de mostrar les preguntes ja contestades.

Per totes aquestes raons, APIS és el projecte base que hem escollit per a desenvolupar la nostra aplicació, ja que ens aporta una estructura intuïtiva i sòlida a més de estar molt encapsulada i amb un aïllament complet entre les diferents parts. Això ens permetrà afegir el nostre servei amb relativa comoditat sense haver de tocar l'estructura interna d'APIS. Per altra banda al ser *open-source* ens permetrà incorporar les últimes modificacions i poder-les adaptar al nostre projecte.

Podem veure una estructura bàsica d'APIS:

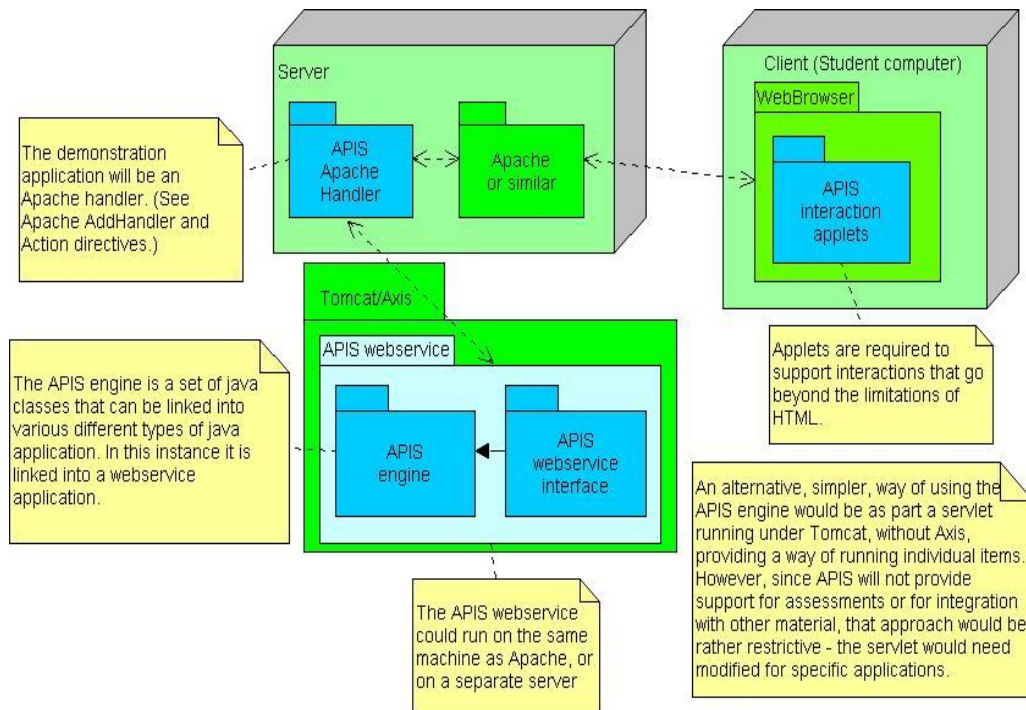


Fig 4: Estructura modular d'APIS

### 5.3 Descripció de la implementació

El projecte d'APIS està implementat en llenguatge de programació Java, en conseqüència segueix una estructura orientada a objectes.

Pel que fa al seu funcionament, consisteix en el següent procés:

Un cop s'ha descrit l'arxiu XML del qüestionari i els seus corresponents *items*, APIS utilitzant la llibreria SAXParser, recupera les dades que hi ha descrites dins l'arxiu, per tant, per cada tag, hi ha una classe que el processa i a partir d'això genera el codi XHTML per a construir la plana web on hi haurà el qüestionari.

L'organització general és la següent:

- *apis*: És el motor d'*items*, com hem dit abans fa la conversió dels *items* definits en l'estàndard IMS QTI (definitos en un XML mitjançant tags) a planes web XHTML.
- *apistesthost*: És on hi ha situat el servidor web i totes les eines necessàries per a llegir i escriure planes web.
- *apis.expression*: És on s'estructura la lògica de les expressions.
- *nb\_utils*: És on es situa l'eina del parser per a llegir fitxers amb tags.
- *apisws*: Conté les eines per generar webservices.
- *apisws.rqp*
- *embeded*

Un cop hem explicat l'estructura general és hora de tenir una visió una mica més específica de com s'organitzen les classes:

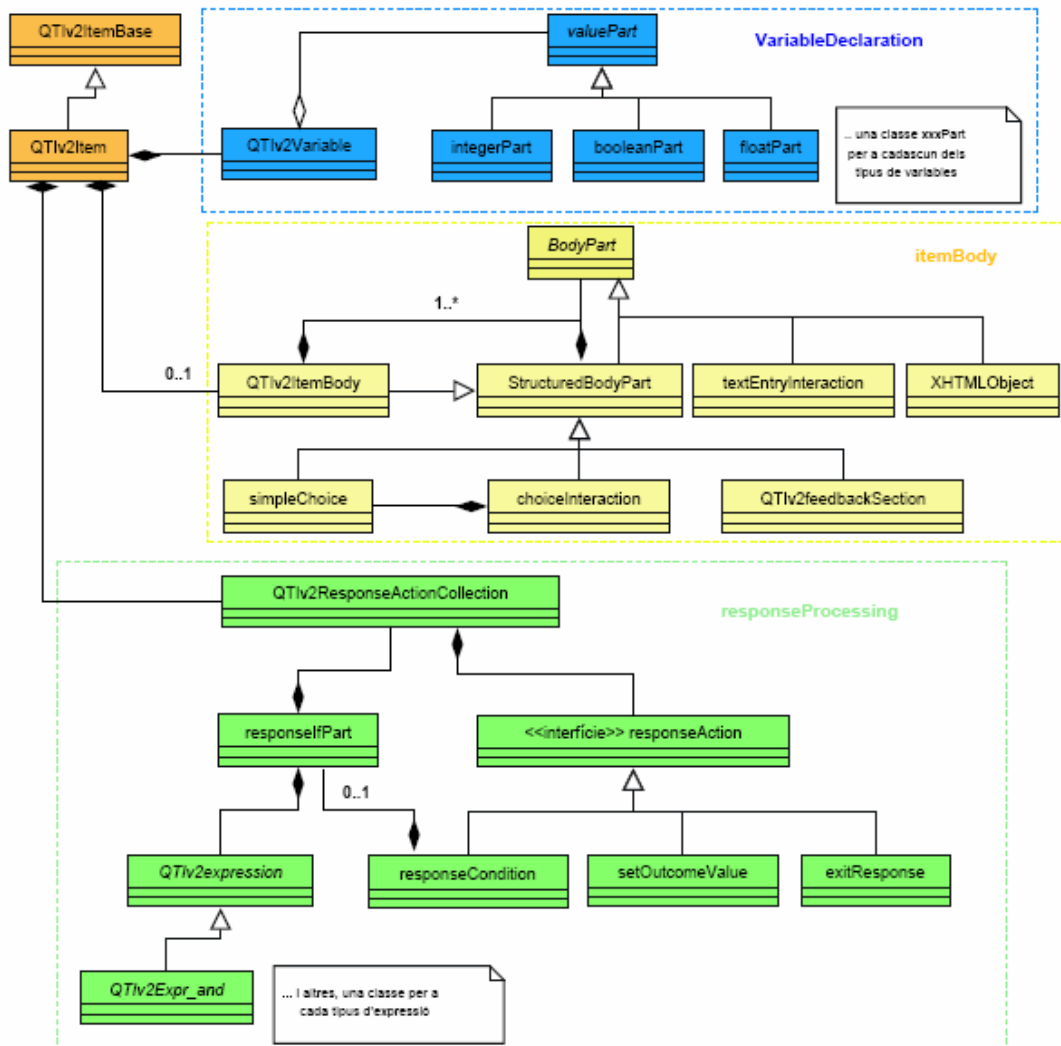


Fig 6.3: Estructura de classes d'APIS

## 5.4 Estat Actual

Com hem dit abans APIS codifica la feina realitzada per IMS i la transforma en una aplicació funcional per a plasmar els avantatges que caracteritzen els esforços d'estandardització.

La situació anterior d'APIS només implementava una petita part de l'especificació IMS Question and Test Interoperability v2.0. Per aquest motiu, encara calia realitzar grans avanços en aquest projecte per a que pogués cobrir tota l'especificació. Alhora també existeix la versió 2.1 de l'especificació que no està contemplada a APIS.

Per a la realització del nostre projecte de final de carrera escollim la versió d'APIS realitzada per Helena Batlle en el seu projecte (Universitat Pompeu Fabra) [28]. En aquesta versió, com hem esmentat en capítols anteriors, hi ha

implementada una gran part de la versió 2.0 i també una part important de la versió 2.1 de IMS QTI que es centra en els tests [28].

En aquesta versió d'APIS estan implementats usos com:

- Lectura del test: Llegir atributs i continguts d'un document XML que defineix el test.
- Validació del test QTI 2.1: Validar els test seguint les normes d'aquesta versió de l'especificació. És tasca del sistema, assegurar-se de que no hi ha errors en la definició com ara elements que no hi haurien de ser.
- Processat: Un cop llegit i feta la validació, s'han de processar les dades per poder reproduir-lo i calcular els valors de les variables de resposta. La correcció de les respostes es realitza de la següent forma: Segons la resposta que seleccioni i el tipus d'interacció, la plana web, genera codi HTML que APIS recull, aquest codi generat el compara amb la resposta correcta que hi ha escrita en la descripció de *l'item*. Si les dues cadenes de caràcters coincideixen, APIS atorga una puntuació determinada, amaga la pregunta que hi havia i mostra la nota en el seu lloc.

Per últim en aquest projecte també es fa un disseny més amigable de les interfícies de resposta per tal de que sigui més comprensible per part de l'alumne.

Degut a que és la versió més actualitzada i per la sèrie d'avantatges abans esmentades que incorpora, la integració del nostre servei es farà en aquesta aplicació.



## 6. El Servei de GoogleMaps

---

### 6.1 Introducció

De manera general tothom amb uns coneixements mínims d'informàtica ha utilitzat GoogleMaps, aquest servei és proporcionat per l'empresa Google el qual ofereix imatges via satèl·lit de tot el planeta combinades amb ortofotografies per a alguns emplaçaments més detallats.

El servei el podem trobar a la plana <http://maps.google.com>, i des d'aquí es pot cercar qualsevol emplaçament del món obtenint una fotografia com a resposta. El format de la cerca es fa de la següent manera: *Carrer, número, població, (ciutat districte, comunitat, etc..)* i *país*. També es pot buscar per diferents continents i accepta les cerques de les ciutats en diferents idiomes, no cal que sigui el nom en la llengua original del país [12]. A nivell d'usuari també fa possible personalitzar una mica els mapes i guardar-los, sempre i quan es disposi d'un compte d'usuari registrat. Amb això, a més, dóna la possibilitat de marcar emplaçaments, dibuixar línies o polígons i emmagatzemar adreces cercades amb anterioritat, mantenint un historial.

### 6.2 Característiques i funcionament del servei

GoogleMaps funciona com un client de visualització cartogràfica. Proporciona una cobertura de nivell mundial (més detallat en emplaçaments importants fins a nivell de carrer). Per altra banda també ofereix la següent informació:

- Cartografia vectorial: És la cartografia representada mitjançant vectors digitalitzats, serveix per a representar els carrers de la ciutat.
- Fotografia satèl·lit: Està codificada com una matriu de punts (igual que la fotografia digital) capturada per un sensor instal·lat a un satèl·lit que orbita la Terra. A mida que el satèl·lit avança en la seva òrbita, va fent fotografies.
- Ortofotografies: Són representacions fotogràfiques d'una zona de la superfície terrestre, en el qual tots els elements tenen una mateixa escala, lliure d'errors i deformacions amb la mateixa validesa que un pla cartogràfic.

Una ortofotografia s'obté mitjançant un conjunt d'imatges aèries (preses des de un avió o satèl·lit) que han estat corregides digitalment per a representar una projecció ortogonal sense efectes de perspectiva i per tant és possible realitzar-hi mesures exactes. A aquest procés de correcció se l'anomena ortorectificació. En resum, una ortofotografia combina les característiques en detall d'una fotografia aèria amb les propietats geomètriques d'un pla.

Veiem l'exemple:



Fig 6.1.1: Vista aèria d'una població amb visió zenital

En zones on el nivell de detall és molt gran es poden arribar a resolucions d'1 o 2 metres per píxel.

GoogleMaps treballa amb escales i mesures específiques que cal conèixer, per tant, cal aclarir alguns conceptes bàsics que s'aniran esmentant durant el desenvolupament del projecte.

Per a representar qualsevol punt de la Terra, es necessiten coordenades esfèriques. Dues d'aquestes coordenades estan expressades en graus sexagesimals i són:

- *latitud*: és la distància que existeix entre un punt qualsevol i l'equador de la Terra. Tots els punts ubicats dins d'un mateix paral·lel tenen la mateixa latitud.
  - Aquells que es troben al nord de l'Equador reben la denominació de Nord (N)
  - Aquells que es troben al Sud de l'Equador tenen denominació Sud(S)
  - Es fa la mesura a partir de 0° fins a 90°
  - A l'Equador li correspon la latitud 0°

- Els pols Nord i Sud tenen latitud  $90^{\circ}\text{N}$  i  $90^{\circ}\text{S}$  respectivament.

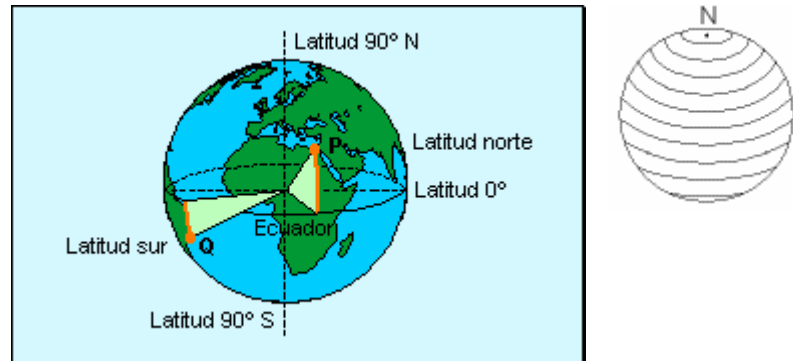


Fig 7.2.2 Latituds de la terra en graus

- *Longitud*: és la distància que existeix entre un punt qualsevol i el Meridià de Greenwich, mesurada sobre el paral·lel que passa per aquest punt.
  - Tots els punts ubicats sobre el mateix meridià tenen la mateixa longitud
  - Aquells que es troben a l'orient del meridià de Greenwich reben la denominació Est(E)
  - Aquells que es troben a occident del meridià de Greenwich reben la denominació d'oest.
  - El rang oscil·la entre  $0^{\circ}$  i  $180^{\circ}$
  - Al meridià de Greenwich li correspon la longitud  $0^{\circ}$
  - L'antemerià corresponent està ubicat a  $180^{\circ}$
  - Els pols Nord i Sud no tenen Longitud.



Amb aquestes dues magnituds es pot representar qualsevol punt a la superfície de la Terra, però si volem representar un punt al cel o sota terra necessitem una tercera dimensió:

- *Alçada*: Per a determinar una posició completament a sota o sobre la superfície de la Terra, s'especificarà mitjançant l'elevació o alçada. Aquesta magnitud pot ser expressada relativa a una dada com per exemple el nivell del mar o la distància al centre de la Terra. En metres o peus. A GoogleMaps però es representa amb un valor propi que oscil·la entre 0 i 23.

El funcionament del servei és el següent : el client fa peticions al servidor d'unes determinades coordenades geogràfiques, segons el nivell de zoom, el mapa es descompost en un número determinat de blocs o quadrats. Aquesta descomposició, que depèn del zoom, es fa automàticament per un *script* a la banda del servidor.

Un cop es fa aquesta descomposició, el mapa es mostra per blocs, de vegades, quan la velocitat de connexió no es gaire ràpida, es pot observar aquest procés, ja que comença a carregar els blocs en ordre aleatori i poden quedar blocs sense carregar.

Quan l'usuari fa lliscar la imatge mostrada en un sentit o altre, el codi del servidor calcula automàticament quins són els quadrats afectats per aquest lliscament. El client torna a fer la petició d'imatges i el servidor retorna les imatges que falten alineant les noves al costat de les antigues. Per tant, per cada moviment del client, no cal tornar a fer una petició de totes les imatges altre cop. Així la velocitat del servei augmenta ostensiblement. Aquest fenomen a banda de l'acció de lliscar també s'aplica a l'opció de zoom, on també s'actualitza el zoom per blocs, però en aquest cas sempre cal que tots els blocs visibles s'actualitzin quan l'apropament canvia.

Cada bloc s'emmagatzema en un fitxer amb un nom que indica la seva longitud, la seva latitud i el seu zoom. Aquestes dades són les imprescindibles per a situar-se en qualsevol punt del globus terrestre (latitud i longitud) amb una determinada alçada de la superfície terrestre (zoom).

Per tal de recuperar aquestes tres dades per a tots els quadrats a col·locar, simplement es realitza una derivació de les dades conegudes per a un sol quadrat cercant les relacions matemàtiques.

Per tant, aquesta tecnologia implementada per Google, necessita una sincronització perfecte entre el que pot preguntar el client i el que efectivament és disponible al servidor de fotografies de la Terra.

Cal incidir en que és molt important que l'algorisme de retall inicial de la imatge completa sigui molt precís i acurat ja que és el responsable, per a la precisió de tall i el seu etiquetatge, dels esdeveniments desencadenats per les accions dels usuaris. Si aquest algorisme no funcionés de manera precisa podria haver duplicació de blocs o omissió de zones a mostrar.

Hem escollit aquest servei de mapes ja que encara que no sigui molt precís (de fet es poden trobar errors d'escala, perspectiva i visibilitats), ens serà una eina molt útil de cara a fer preguntes de geografia, ja que alhora que és molt popular i familiar en l'ús; té una gran accessibilitat i és intuïtiu.

Una altra característica fonamental es que GoogleMaps, com altres aplicacions de Google, utilitza com a base JavaScript, per tant serà la tecnologia que utilitzarem per a la implementació i l'accés als mapes alhora d'elaborar els qüestionaris.

## 6.3 JavaScript

El llenguatge va ser inventat per Brendan Eich a l'empresa Netscape Communications que és la que va fabricar els primers navegadors web comercials. Aparegué per primera vegada en el Netscape Navigator 2.0

JavaScript és un llenguatge interpretat, utilitzat normalment en pàgines web, té una sintaxi semblant a la del llenguatge Java i el llenguatge C.

Al contrari de Java, JavaScript no és un llenguatge orientat a objectes en essència ja que no disposa d'herència. JavaScript és un llenguatge basat en prototips ja que les noves classes generen clons de les classes base (prototips) i estenen la seva funcionalitat.

Tots els navegadors interpreten JavaScript integrat dins el codi de la pàgina web. Per a interactuar amb una pàgina web es proveeix al llenguatge JavaScript una implementació del DOM.

El codi es pot incloure en qualsevol document d'HTML o tot aquell que es tradueixi en codi HTML com PHP, ASP, SVG... Es pot inscriure de la següent manera:

```
<script type="text/javascript">  
    // codi JavaScript  
</script>
```

Dins d'aquest codi es poden dissenyar algorismes o funcions que resolguin una sèrie d'operacions que després poden generar codi HTML amb els resultats.

## 6.4 GoogleMaps com a eina d'aprenentatge

GoogleMaps no va sorgir amb l'objectiu de ser una eina per a ser emprada de cara a l'aprenentatge; pel contrari només pretenia ser un servei per a satisfer curiositats, donar una visió global del món i poder aplicar mesures amb una certa exactitud.

Degut a la quantitat de possibilitats i capacitat de personalització que ofereix creiem que en un context adequat pot ser una eina útil per a l'e-Learning. Al tenir una API pública, com ja hem comentat abans, ens permet utilitzar només una part de totes les possibilitats que ofereix i adaptar-les per a satisfer els objectius del nostre projecte. Principalment hem escollit aquest servei per les següents capacitats i característiques:

- *Cercador*: Fa possible, d'una manera fàcil i ràpida, visualitzar qualsevol punt de la Terra només introduint un topònim o carrer concret sense ambigüitats. Des del punt de vista educatiu, un professor podrà cercar el mapa que vol mostrar d'una manera lleugera, còmode i ràpida.
- *Situar un mapa*: Amb aquest servei també es pots posicionar a qualsevol punt de la Terra sabent les seves geocoordenades exactes. De cara al professor podrà generar mapes definint personalment la latitud, la longitud i el zoom.
- *Determinar els tipus de controls*: GoogleMaps té una sèrie de controls sobre el mapa. Mitjançant l'API es poden posar a disposició de l'usuari

eines com: Fer zoom, fer zoom amb la roda del ratolí (més interactiu i intuïtiu), escollir els tipus de mapes (físic, carrers o híbrid), dibuixar, emmagatzemar punts, etc. Amb això podem determinar quines possibilitats té l'alumne alhora de conèixer l'entorn i com interactuar amb ell.

- *Dibuixar geometries:* GoogleMaps té la possibilitat de poder dibuixar directament a sobre del mapa figures geomètriques (punts – marcadors -, polígons, línies) amb el ratolí que en l'àmbit pedagògic pot donar molt marge de maniobra alhora de definir tipus d'interaccions.
- *Control d'esdeveniments:* GoogleMaps defineix uns controladors d'esdeveniments que ens donen la possibilitat de controlar qualsevol acció que es faci a sobre del mapa (click, doble-click, arrossegat, botó dret) i sobre els marcadors. Això també ens dona la possibilitat juntament amb el punt anterior, de definir tipus de preguntes i interaccions molt flexibles i originals que després desenvoluparem.

## 6.5 L'API de GoogleMaps

Des de 2005 GoogleMaps va publicar la seva API per aprofitar les dades disponibles a través del servei. Aquesta API, que està implementada en Javascript, dona la possibilitat de posar a qualsevol pàgina web un mapa personalitzat amb les característiques que vulguem (sempre que ho permeti l'API i el Servei).

Ja que ens proporciona un conjunt de possibilitats i interaccions originals i molt versàtils cal aprofitar-les en l'àmbit educatiu.

L'API de GoogleMaps la podem trobar en aquesta adreça web: <http://www.google.com/apis/maps/> on hi ha un petit tutorial, i a més de la seva referència a totes les classes i funcionalitats: <http://www.google.com/apis/maps/documentation/reference.html>.

Tot seguit mirem l'API en més detall:

### 6.5.1 Estructura de l'API.

L'API de GoogleMaps conté moltes funcionalitats per a cobrir diferents necessitats de diversos tipus de perfils d'usuari, nosaltres explicarem les més importants i les que ens seran útils en el nostre projecte, actualment està en vigor la versió 2.0 d'aquesta [13][14]:

**GMap2:** És la classe principal de l'API, crea un nou mapa dins de la plana HTML on està *l'script*. Des de aquesta classe desenvoluparem totes les funcionalitats que hi haurà a sobre del mapa.

*enableDragging()*, *disableDragging*

Fa possible que el mapa es pugui desplaçar o no, és útil per a restringir la informació que pugui tenir l'alumne.

*enableScrollWheelZoom()*

Permet fer zoom amb la roda del Mouse.

*addControl(control, position?)*

Són tipus de controls, com per exemple botons per seleccionar el tipus de mapa o si volem que hi hagi una barra de control de zoom.

*addOverlay(overlay)*

Els overlays són informació o capes superposades al mapa que ofereixen informació personal i addicional.

*getCenter(), getBounds, getSize(), getZoom()*

Aquestes funcions són útils ja que ens permet recuperar informació associada al mapa.

*mouseover(), mouseout(), mousemove(), dragstart(), dragend()*

Aquestes funcions permeten llençar esdeveniments quan hi ha alguna d'aquestes accions.

*setCenter(center, zoom?, type?)*

Fa possible definir el centre on es situa el mapa, el nivell d'alçada i el tipus de mapa.

Els tipus de mapa que podem posar són els següents:

- **G\_NORMAL\_MAP**: És el mapa original, mostra la informació dels carrers.
- **G\_SATELLITE\_MAP**: Mostra els mapes de Google Earth de tipus satèl·lit.
- **G\_HYBRID\_MAP**: Mostra els carrers de manera transparent a sobre dels mapes tipus satèl·lit.

**GMarker**: Marca una posició al mapa. Implementa GOverlay, és a dir, funciona com un Overlay. El marcador es situa a una coordenada en el mapa i una icona (té una per defecte però es poden posar personalitzades).

*openInfoWindow()*

Obre una bafarada (blowUp) amb informació personalitzada, pot tenir diferents pestanyes i mostrar un altre mapa o text html.

*enableDragging() i disableDragging()*

Fa possible que el GMarker es pugui arrossegar.

*Click(), doubleClick(), ....*

Dóna la possibilitat de detectar una sèrie d'esdeveniments a sobre del marcador.

**GPolyLine** (*points, color?, weight?, opacity?*):

Fa possible crear línies definides per una sèrie de punts, amb un determinat color, una determinada opacitat i una determinada amplada

**GPolygon:** Permet crear polígons de la mateixa manera que les línies, l'àrea definida pel polígon es pot ressenyar amb un color.

**GLatLng:** És la classe que defineix un punt a sobre del mapa amb coordenades geogràfiques, latitud i longitud.

*Lat(), lng()*

Fa possible accedir separadament a les coordenades.

*distanceFrom(other)*

Retorna la distància entre un punt geogràfic i un altre.

**GPoint:** Funciona igual que GLatLng però retorna un punt x,y en els quals els píxels de la imatges són les coordenades.

**GEvent:** Aquest *namespace* conté funcions que es poden utilitzar per a capturar esdeveniments sobre el mapa.

*addListener(source, event, handler)*

Escolta l'esdeveniment definit i la funció que ha de fer quan succeeix.

**GGeoXml:** Dóna la possibilitat d' afegir overlays i arxius kml i kmz



## **DESENVOLUPAMENT DEL PROJECTE**



## 7. Anàlisi dels Requeriments

---

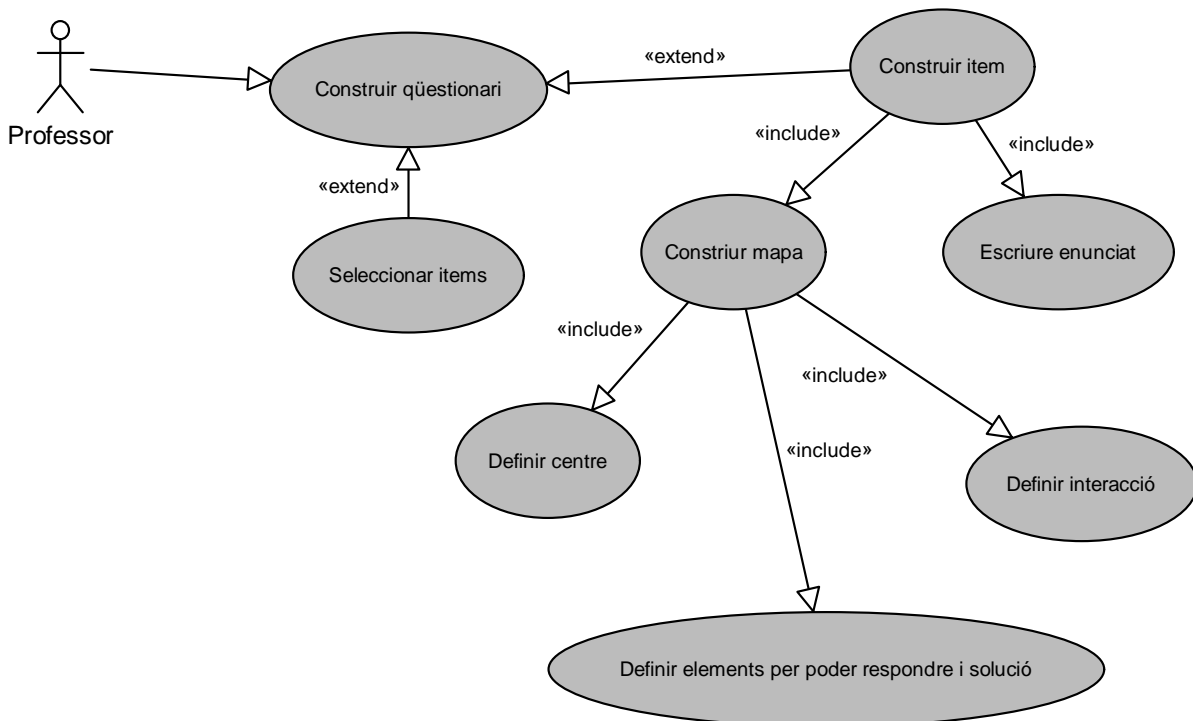
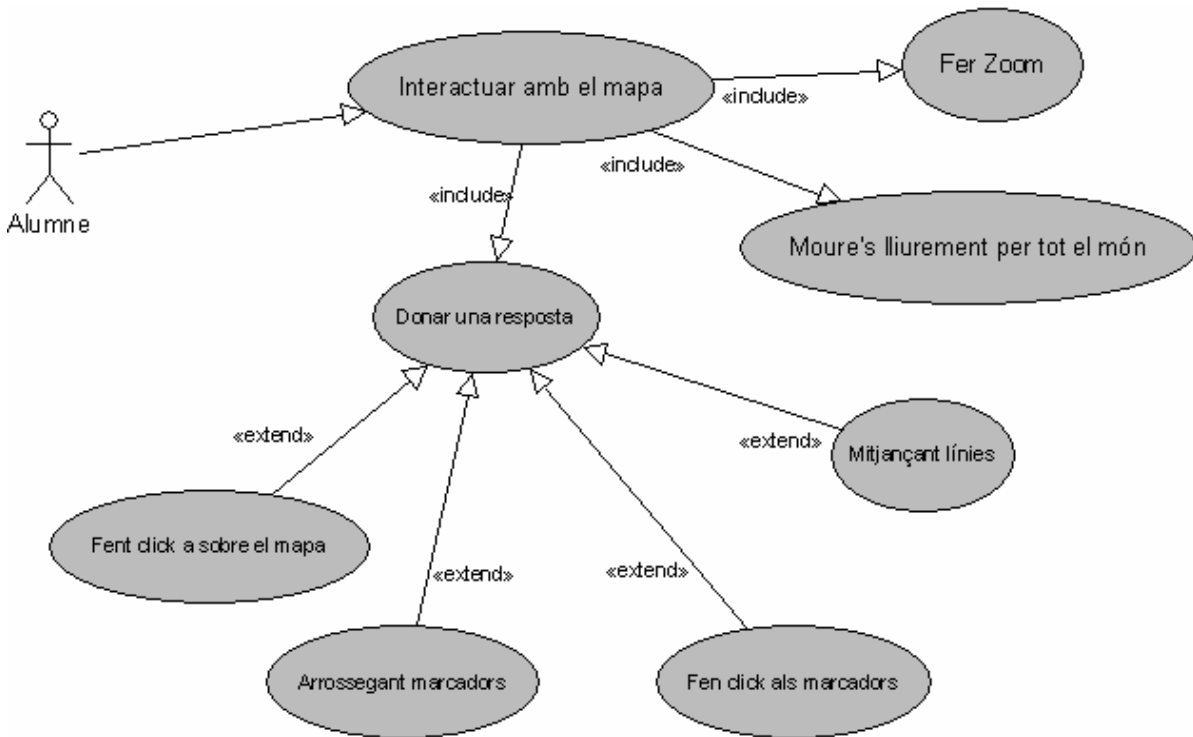
Després d'explicar els objectius i contextualitzar l'àmbit en el qual es mourà el projecte explicant les diferents tecnologies utilitzades, cal ara fer una bona anàlisi de requeriments per tal de definir un correcte camí per a desenvolupar el projecte.

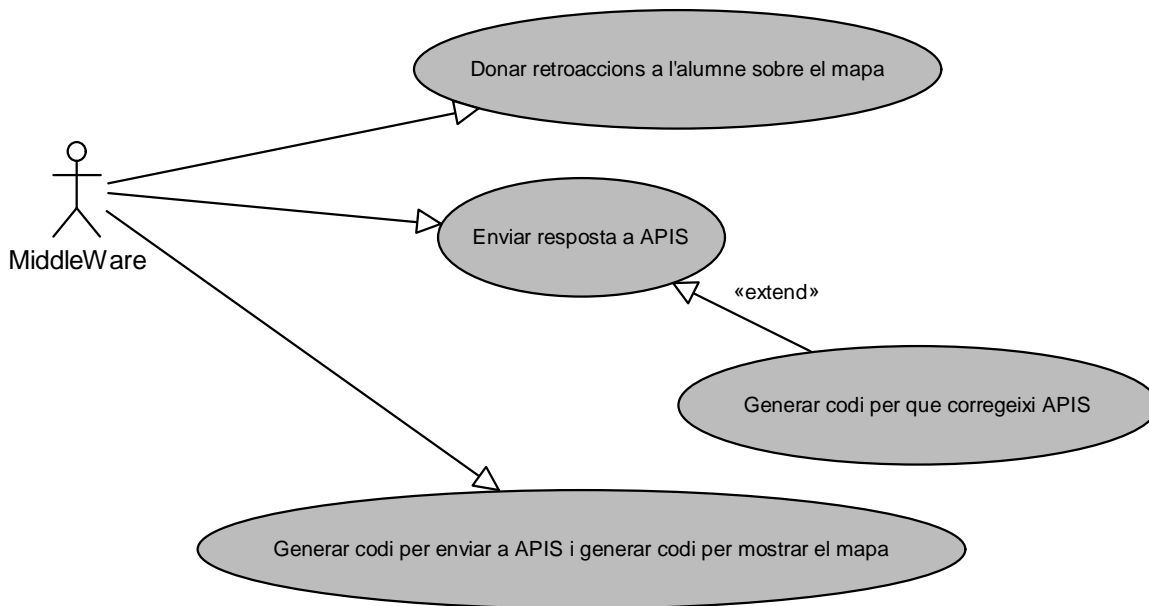
Els punts fonamentals on es recolza la nostra solució són els següents:

- Connexió entre els dos *engines*: Tant APIS com GoogleMaps són dos sistemes independents que funcionen per separat, per tant, per a poder fer que treballin conjuntament haurem de dissenyar una estructura per a poder connectar ambdues aplicacions.
- Disseny del middleware: Cal que el middleware satisfaci les necessitats de comunicació. Per tant haurà de traspasar les dades de manera eficient i integradora. El middleware adaptarà la resposta de GoogleMaps a un codi que pugui ser integrat per APIS en una plana web.
- Disseny d'una estructura de dades per a representar dades Geoespaciales: un mapa de GoogleMaps conté una sèrie de característiques que han de poder se representades mitjançant un format determinat dins d'una estructura XML. Aquesta estructura haurà de complir un disseny jeràrquic, intuïtiu i de fàcil comprensió per a que sigui còmode poder generar *ítems*.
- Disseny d'interaccions: cal estudiar des de el punt de vista de l'alumne i l'avaluador quins tipus d'interaccions són d'utilitat per al procés d'aprenentatge de la Geografia explotant al màxim i de maneres originals totes les eines que ens ofereix l'API de GoogleMaps.
- Validació de les respostes i correcció: Per finalitzar, cal que el middleware generi el mateix codi de correcció que genera APIS per a poder enviar-li la resposta i que APIS pugui fer la correcció, comparant la resposta correcta definida a l'*ítem* amb la que li traspassa el middleware.

Un cop hem vist de manera general quins estudis, dissenys, funcionalitats i implementacions cal abordar, descriurem més detalladament mitjançant els casos d'ús, els requeriments funcionals, no funcionals i d'usuari les característiques del nostre projecte.

## 7.1 Diagrama de casos d'ús





## 7.2 Requeriments funcionals

En aquest punt s'enumeren les funcionalitats que haurà d'aportar el sistema. De manera més detallada es descriuen possibles *inputs* dins del sistema i com reacciona davant d'ells amb les possibles sortides que podrà tenir.

Els requeriments funcionals són els següents:

- La interfície aportada pel servei GoogleMaps, per a poder respondre a les qüestions, haurà de ser comprensible i intuïtiva, al ser preguntes de Geografia, la situació del mapa i el seu nivell de zoom així com la informació mostrada haurà d'estar degudament situada per a complir el seu objectiu pedagògic.
- Aquesta intuïtivitat haurà d'estar complementada amb una bona redacció i situació dels enunciats, és a dir, la interfície generada per APIS que ja se suposa que hi és. El mapa s'ha de situar en un punt dominant de la pregunta ja que ara es converteix en l'eix d'aquesta. Així facilitem la correcta comprensió de la pregunta i una correcta situació visual dels elements mostrats.
- El sistema cobrirà una part de les interaccions definides per l'estàndard QTI en relació a les imatges a tall d'exemple fent un disseny que afavoreixi la incorporació de la resta de l'especificació en aquest àmbit.
- S'hauran d'aprofitar els serveis que proporciona l'API de GoogleMaps per a poder aportar nous models d'interaccions no contemplats a l'estàndard IMS QTI com el zoom, la mobilitat del mapa o les capes semitransparents.

- Les interaccions dissenyades hauran de ser útils des de el punt de vista pedagògic.
- Hi haurà dos tipus d'elements mostrats a sobre el mapa: els exclusivament utilitzats per a poder donar una resposta a la qüestió i els contextualitzadors o d'informació addicional (rius, fronteres, ...).
- El sistema oferirà informació de les accions que està fent l'usuari, és a dir, quina resposta a seleccionat, l'ordre de les seves respostes, on ha clickat a sobre del mapa que siguin clarificadores per a l'alumne.
- El sistema haurà d'extreure les dades del mapa d'un XML amb una estructura preestablerta, interpretar-lo i enviar-li a APIS perquè ho mostri per pantalla.
- La correcció de les respostes les tractarà APIS de manera independent sense fer crides al nostre servei.
- El sistema haurà de ser capaç de reportar, tractar i evitar els possibles errors tant en l'accés a les dades del mapa com en el procés de la interacció.
- El sistema només mostrarà els mapes que encara queden per respondre i amagar els que ja han estat contestats.
- El sistema generarà codi XHTML i HTML per a enviar la resposta.

### 7.3 Requeriments no funcionals

En la implementació del projecte, volem adaptar el nostre servei a APIS, per tant s'imposen una sèrie de restriccions formals i de llenguatge específics que tot seguit comentarem:

- Es construirà un Middleware que es comunicarà amb APIS.
- Com que APIS està implementat en llenguatge Java, per a una fàcil i còmode adaptació, el nostre sistema haurà d'estar implementat també en Java, paradigma de la programació orientada a objectes.
- El nostre Middleware combinarà llenguatge Java, llenguatge JavaScript i llenguatge XHTML generat per JavaScript.
- El nostre Middleware utilitzarà la tecnologia SAXParser per a obtenir les dades del mapa des de un XML que ja utilitza el motor APIS.
- Per a seguir amb el model de IMS QTI, la definició dels elements que hi haurà en el mapa, així com les seves característiques i situacions estarà definit en XML.

- L'estructura i implementació d'APIS haurà de modificar-se el mínim possible.
- Per a que un XML de mapa sigui vàlid haurà de contenir com a mínim la definició d'una interacció i un centre (latitud, longitud i zoom).

## 7.4 Requeriments d'usuari

En aquest punt s'exposaran les funcionalitats que podrà fer l'usuari, en un principi hi hauran diverses interaccions. Entenem per usuari un alumne que haurà de trobar una solució i respondre les preguntes formulades. És necessari trobar una solució a les necessitats de l'usuari per a poder saber quines dificultats i facilitats troba, i així trobar les millors interaccions possibles (fàcils i intuïtives).

- L'usuari haurà de poder contestar a la qüestió proposada a sobre mateix del mapa del servei GoogleMaps.
- L'usuari tindrà la llibertat de moure's per tot el mapa sempre que la qüestió ho permeti.
- L'usuari sempre podrà modificar el zoom i accedir a diferents perspectives i alçades per a veure el mapa. Sempre hi haurà la possibilitat de poder-lo desactivar i restringir si la pregunta ho exigeix.
- El docent o dissenyador del qüestionari dels test podrà decidir la interacció a fer i el tipus de pregunta.
- El docent o dissenyador del qüestionari podrà seleccionar els elements i overlays que hi haurà a sobre del mapa.
- El docent o dissenyador podrà decidir on vol centrar el mapa.
- El docent dissenyarà per separat l'XML de *l'ítem* de la pregunta i l'XML que defineix el mapa.
- El docent no haurà de notar cap canvi alhora de inserir *ítems* que contenen mapes en un qüestionari. Per tant els qüestionaris es definiran de la mateixa manera que en versions anteriors.
- L'estructura de dades del fitxer que contingui les dades sobre el mapa haurà de ser intuïtiva i ordenada, de fàcil comprensió i jeràrquica.

Aquestes, en definitiva, seran les condicions en les quals contestarà l'usuari. Com que no és un projecte tancat, sempre, tots aquests punts podran veure's modificats, millorats o eliminats.





## 8. Disseny i Implementació del Projecte

---

En aquest punt veurem les solucions a nivell de disseny i implementació del nostre projecte. Anirem argumentant les decisions preses en cada moment i el motiu de la solució seleccionada.

### 8.1 El Middleware

En aquest punt veurem com hem implementat el middleware entre el servei de GoogleMaps i APIS.

#### 8.1.1 El middleware com a solució

Implementar un middleware entre dues aplicacions que funcionen independentment per separat esdevé la millor solució per a realitzar aquest procés de comunicació i treball en conjunt per així poder obtenir una nova aplicació mixta resultant de les dues anteriors.

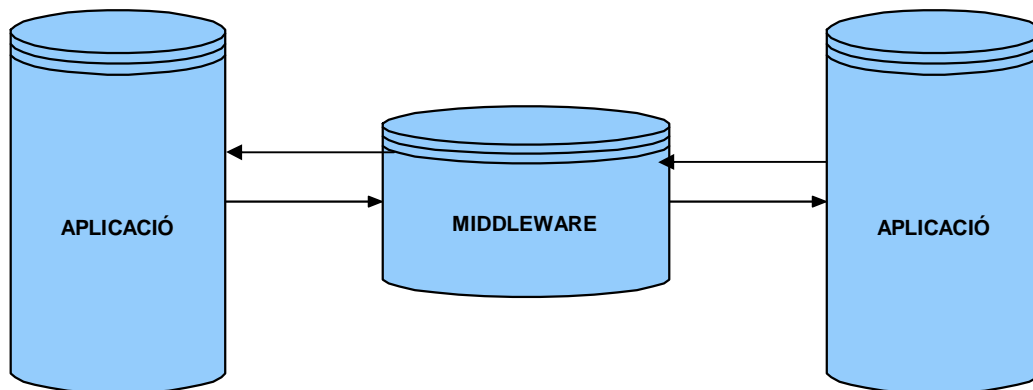


Fig 8.1.1 Situació i direcció de les comunicacions d'un middleware

Les raons per escollir el disseny d'un middleware són les següents:

- *Independència:* Les aplicacions implicades només han de saber comunicar-se amb el middleware i ens dona la possibilitat de traspasar i gestionar dades entre una aplicació i l'altre. D'aquesta manera podem dir que el middleware no pertany a cap de les dues altres estructures
- *Traducció:* Capacitat per adaptar les dades i traduir-les per a la correcte comprensió.
- *Abstracció:* Simplifica el procés de desenvolupament ja que no cal conèixer en profunditat el funcionament intern de les aplicacions connectades.
- *Flexibilitat:* Amb un middleware incrustat enmig de les dues aplicacions com APIS i GoogleMaps podem suportar creixements (com nous tipus de pregunta, noves interaccions i fins i tot una nova estructura) sense haver de modificar les altres dues aplicacions.

- *Transparència:* Degut a la transparència d'un middleware, el creixement i les modificacions són fàcilment realitzables.
- *Seguretat:* Al tenir un middleware, la informació està millor encapsulada i encriptada, ja que un middleware es pot acabar convertint en una API.
- *Diferents ambients de disseny, programació i computació:* Els middlewares ens permeten fer entendre dins d'un mateix espai, estructures de programació i disseny diferents, fins i tot diversos llenguatges de programació.

### 8.1.2 Objectius i Disseny

El disseny d'un bon middleware que comuniqui de manera bidireccional APIS i el servei aportat per GoogleMaps esdevé de capital importància.

Aquest middleware serà un paquet que inclou diferents classes on més endavant veurem la seva organització.

Per a fer la comunicació més eficient hem de veure quin tipus d'informació necessita APIS per a construir els qüestionaris.

Una explicació detallada però d'alt nivell de com funciona APIS seria la següent:

A APIS, quan es selecciona un fitxer on hi ha definits qüestionaris, el que fa és cridar a la seva eina SAXParser que llegeix i recull les dades definides en un document XML que conté l'estructura d'un qüestionari amb una interacció concreta. Aquesta estructura està organitzada en tags que segueixen l'especificació IMS Question and Test Interoperability.

Aquest procés el realitza per a tots els *items* que hi hagi creats. Un cop ha llegit el document, va recollint la informació, i per a cada tag que implementa l'especificació IMS QTI es crida a una classe que hi ha implementada a APIS que correspon amb aquest tag. Dins d'aquesta classe es genera codi HTML que es va afegint a una variable que emmagatzema cadenes de caràcters.

Un cop s'ha generat el codi per a llençar el formulari amb la pregunta a realitzar i el codi necessari per a processar la resposta dins de la variable content, APIS escriu aquesta cadena de caràcters en un fitxer HTML el qual és interpretat per un navegador web.

Més esquemàticament ho podem veure en el següent diagrama:

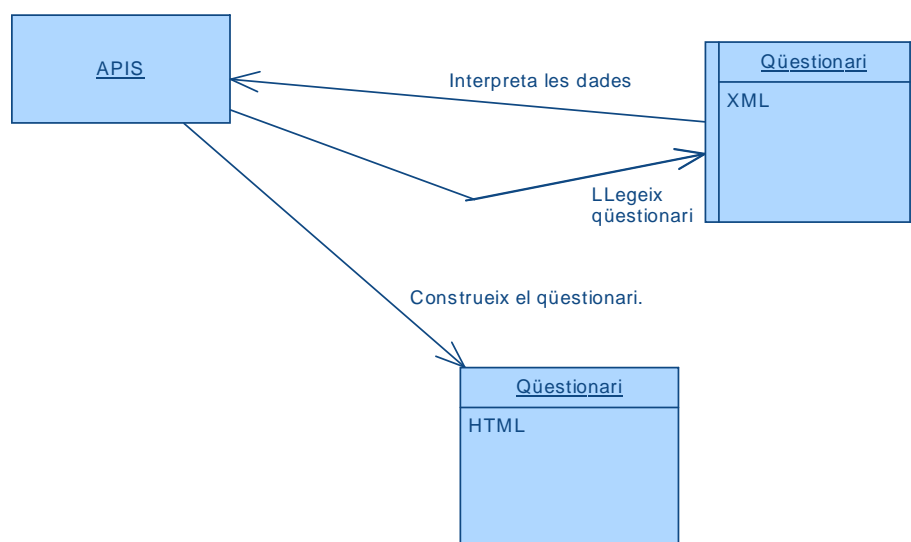


Fig 8.1.2.1 Funcionament d'APIS

Per altra banda el servei de GoogleMaps funciona de tal manera que es van fent crides a la seva API mitjançant codi en JavaScript, aquest codi en JavaScript ha d'estar incrustat dins d'una pàgina HTML. A mida que es criden les funcions de l'API aquestes retornen dades que permeten construir un mapa amb diferents opcions.

La funcionalitat i la comunicació esdevé de la següent manera:

Volem que les preguntes puguin mostrar mapes de GoogleMaps on es pugui interactuar a sobre (a la fase d'implementació explicarem com s'ha fet). En conseqüència cal definir com fer que APIS detecti que en una zona de *l'item* es vol mostrar un mapa. Un cop s'ha detectat que APIS vol un mapa, la primera comunicació serà APIS-> Middleware.

En segon lloc, el middleware rep una petició per mostrar un mapa amb unes determinades dades de posicionament i elements; llavors el middleware recollirà aquestes dades d'un arxiu XML i un cop les emmagatzema, anirà fent les crides a l'API de GoogleMaps amb aquestes dades per així obtenir el mapa personalitzat que ha demanat APIS. En conclusió, la segona comunicació serà des de el middleware a l'API de GoogleMaps. Aquesta API es comunicarà amb el seu servei privat i ens anirà retornant les dades que el middleware recollirà.

Entrant en més detall en aquest aspecte no és del tot correcte aquesta apreciació, ja que les crides a l'API de GoogleMaps s'han de fer des de una plana web en activitat, per tant el que realment fa el middleware, és generar el codi necessari en JavaScript per a que després quan es carregui la plana cridi a l'API de GoogleMaps i pugui generar el mapa. Per tant la següent comunicació és el middleware amb APIS altre cop, retornant-li aquest codi necessari. Per últim quan APIS llenci al navegador la seva plana amb el qüestionari, també escriurà el codi en JavaScript que farà les crides al servei GoogleMaps.

Més gràficament la connexió bidireccional serà la següent:

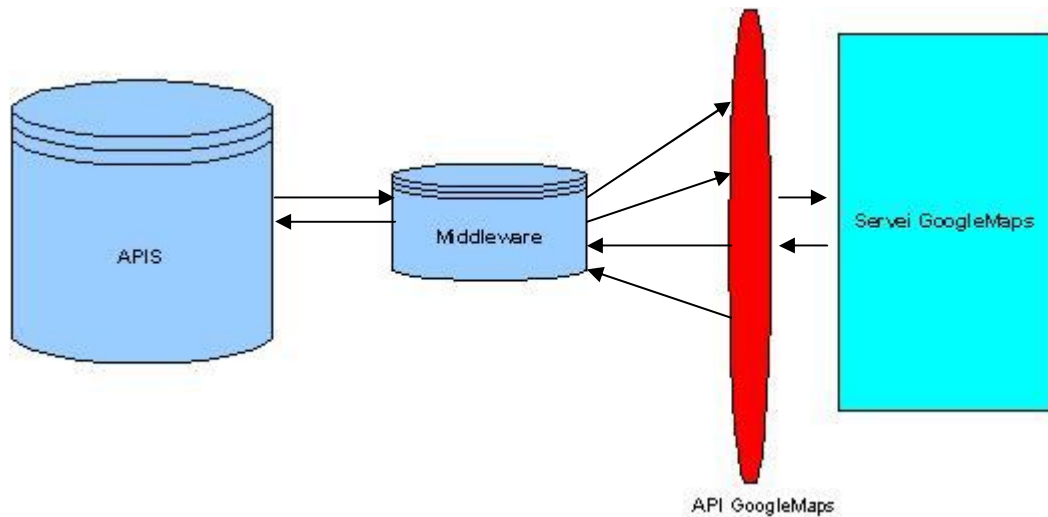


Fig 8.1.2.2 Situació del nostre middleware respecte les aplicacions

### 8.1.3 Implementació de la connexió entre APIS i el MiddleWare

El primer pas alhora de posar en funcionament el nostre servei és fer la connexió entre APIS i el middleware per fer possible la comunicació entre els dos mòduls.

La implementació té per objectiu no modificar en gran mesura el codi d'APIS, d'aquesta manera es va descartar des de el primer moment fer una nova classe per a generar un tipus de pregunta amb GoogleMaps.

Llavors calia resoldre en quin moment en l'execució del codi d'APIS s'hauria de realitzar la connexió amb el middleware i fer la petició d'un mapa.

Hi havia moltes alternatives però es va decidir que per a mantenir la coherència amb l'estructura de dades i els tipus d'elements que estaven implementats era bo tractar un mapa ofert pel servei de GoogleMaps en el mateix moment que es tracta una imatge. Per tant el primer pas va ser fer una anàlisi de la classe i els mètodes on es tracten les imatges:

Quan el docent o qualsevol altre tipus d'usuari vol que aparegui una imatge estàtica contextualitzant la pregunta en un *item* cal que APIS la reconegui.

Per tal de que això sigui així cal afegir al document XML que defineix l'*item* la següent estructura dins del tag `<itemBody></itemBody>`:

```
<p>  
    
</p>
```

Com podem veure al crear qualsevol pregunta en format XML seguint l'especificació QTI també podem incloure tags d'HTML, per tant quan APIS detecta que hi ha una imatge mitjançant aquesta línia, es crida directament a la classe `QTIv2XHTMLObject.java` la qual processa tots els tags d'XHTML.

Així, com hem dit abans, volem tractar les imatges de GoogleMaps en el mateix lloc on es processen les imatges normals, però en comptes de passar-li un fitxer d'imatge, se li passarà un document XML. Per tant la primera modificació feta en la classe abans esmentada és la següent:

```
Si el tag detectat es un XHTML_img
{
    Llegim l'extensió del fitxer que s'ha posat en l'XML que defineix la
    pregunta
    Si el fitxer acaba en .xml
    {
        Crida al middleware per tal d'obtenir les dades del mapa i crear el
        codi per a muntar un mapa ofert per Googlemaps
    }
    else Processa com una imatge normal
}
}
```

La connexió es realitza més detalladament en el moment que cridem des de aquest mètode al middleware, el middleware generarà codi en JavaScript que retornarà a la variable *content* (que és on guarda tot el codi que genera a partir dels tags definits en l'XML de l'ítem). D'aquesta manera a partir del moment que es crida a la funció APIS delega al middleware que funcionarà amb total independència.

Aquestes són les línies en el mètode que detecta les imatges per a delegar al middleware:

```
GoogleMaps map = new GoogleMaps();
content = map.pintarMapa("/items/"+src);
```

On *pintarMapa* retorna l'String amb el codi que generarà el middleware del mapa en JavaScript.

En definitiva, d'aquesta decisió es desprenen dues conseqüències i requisits:

- a) Quan vulguem que aparegui un *GoogleMaps* en comptes d'una imatge li passarem un fitxer de "text amb les dades".
- b) Per tal de que sigui un mapa, haurà de ser un arxiu amb extensió XML i que contingui l'estructura de les dades del mapa que en el següent punt explicarem.

Un cop hem vist que la connexió és possible mitjançant el tag de la imatge, un disseny més acurat des del punt de vista lògic és implementar APIS per tal de que reconegui una altra etiqueta, en aquest cas el tag *map*, que també serà del tipus *QTIv2XHTMLObject*. Aquesta nova etiqueta serà reconeguda per APIS com *XHTML\_map* i funcionarà de la mateixa manera que abans, l'única diferència és que aquest tipus sempre esperarà xml's, en canvi a l'anterior forma d'implementar-lo havíem de controlar *parsejant* si acabava en extensió .xml o qualsevol format d'imatge. Per tant la definició dels mapes en aquesta nova implementació quedarà de la següent manera:

```
<p>
  <map src="GoogleMaps4.xml"/>
</p>
```

### 8.1.4 Implementació del MiddleWare i connexió amb GoogleMaps

Com dèiem en un altre dels requeriments, la codificació del middleware es fa en llenguatge Java per així fer possible una fàcil i correcte adaptació.

El middleware per definició no pertany a APIS, per tant el que s'ha dissenyat és un paquet (*package*) on estaran implementades totes les classes que necessitarem per tal de que funcioni el nostre pont de comunicació entre APIS i GoogleMaps.

La primera classe que s'implementa és l'anomenada *GoogleMaps.java* que té per objectiu generar el codi (segons els elements que tingui i que després explicarem) que crea un mapa i li envia a APIS mitjançant una cadena de text.

Un cop definida la classe principal del nostre middleware, farem una petita explicació de com funciona a nivell de disseny i implementació encetant les decisions que s'han pres.

La classe incorpora alguns atributs com:

- *Latitude* (latitud): Que ens servirà per guardar localment la latitud del centre del mapa. La latitud es guarda en decimal de coma flotant.
- *Longitud* (length): Serveix pel mateix objectiu que la latitud. Serà del mateix tipus.
- *Zoom*: En servirà per a guardar el zoom. En aquest cas es guardarà com a tipus sencer ja que com els mapes seran utilitzats amb finalitats pedagògiques la precisió en aquest camp no té gaire interès.

Aquests tres atributs només serviran com a control, és a dir, si aquests tres atributs no existeixen vol dir que l'arxiu XML amb les dades del mapa no està correctament definit (no hi ha situat un centre on focalitzar el mapa). En conclusió el mapa no es podrà pintar i s'haurà de llençar un error.

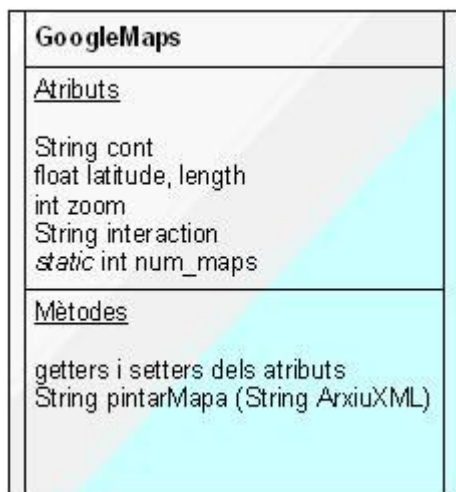
Per altra banda tindrem altres atributs que seran més importants.

- *Interaction* (Tipus d'interacció): Servirà per a recuperar de l'arxiu XML la interacció sobre el mapa que es vol realitzar. Això és degut, com més endavant explicarem, a que per a una mateixa configuració d'elements a sobre del mapa es poden realitzar interaccions diferents.
- *num\_maps*: Aquest atribut és estàtic, és a dir, que aquesta variable és compartida per totes les instàncies de la classe. Aquesta variable s'ha implementat degut a un problema amb els *divs*, ja que alhora de generar el processat de les respostes cal controlar sobre quin mapa s'està responent. És de tipus sencer.
- *Cont* (contingut): Aquesta és una de les variables més importants, ja que funciona de manera semblant a la variable *content* d'APIS, per tant emmagatzema i va afegint les cadenes de text amb el codi JavaScript que crida al servei de GoogleMaps i genera el mapa.

Els mètodes que hi ha implementats són evidentment tots els getters i els setters per a l'accés i modificació de tots els atributs de la classe que fa de pont entre APIS i GoogleMaps.

El mètode principal de la classe és l'anomenat *pintarMapa*, el qual rep com a paràmetre un String, que és el nom del fitxer que li envia APIS amb les dades en format XML del mapa, i retorna un altre String que és el codi en JavaScript que generarà el mapa en la plana web construïda per APIS.

Veiem de manera formal la classe:



L'algorisme d'aquesta classe és senzill, partim de la següent premissa:

Quan un professor vulgui posar un mapa de GoogleMaps en un ítem voldrà variar el tipus de pregunta i la interacció segons les seves necessitats, a més de modificar la informació i feedbacks que retorni el mapa; també els elements que hi haurà a sobre d'ell seran diferents.

Per aquest motiu, el middleware haurà de afegir unes línies o altres de codi segons les característiques definides a l'XML del mapa.

D'aquesta manera per a construir el mapa, cal tenir clar quins elements possibles totals podrà suportar el mapa i així segons els elements que hi hagin es podrà anar afegint el codi adient per que després sigui mostrat a la plana web.

En resum, depenent de les característiques del mapa que s'hagin definit en l'XML (el del mapa) s'aniran afegint trossos de codi que implementen i fan les crides a l'API de GoogleMaps necessàries per a satisfer les necessitats de la pregunta que vol el docent.

L'algorisme bàsic és el següent:

```
String pintarMapa (String arxiuXML)
{
    Creem un SAXParser
    Llegim mitjançant el parser l'arxiu "arxiuXML"
    Generem una estructura de dades amb tots els elements possibles que poden haver-hi

    Per a tots els elements possibles que pugin existir en el mapa fes
    {
        Si existeix l'element (i) llavors
        {
            genera el codi per a mostrar aquest element i fer possible la seva correcció
            cont+= "codi en JavaScript"
        }
    }
    return cont; // es retorna l'string amb tot el codi concatenat
}
```

## 8.2 Representació de les dades de GoogleMaps

En aquest apartat veurem l'estructura del middleware i com s'anirà generant i omplint les dades de l'estructura a mida que el SAXParser vagi recollint les dades. Primer de tot però, cal definir els elements de l'arxiu XML i després s'explicarà com traduir aquesta estructura XML en classes i atributs.

### 8.2.1 Codificar dades en XML (eXtensible Markup Language)

XML es proposa com un estàndard per a l'intercanvi d'informació estructurada entre diferents plataformes. Es pot utilitzar en bases de dades, editors de text, fulls de càlcul i en qualsevol tipus de procediment o aplicació informàtica imaginable. A banda d'això, XML fa possible la compatibilitat entre sistemes per a compartir la informació d'una manera segura, fiable i fàcil.

Els principals avantatges de definir les dades amb una estructura XML són les següents:

- És extensible: una vegada dissenyat un llenguatge i s'ha posat en producció és possible estendre'l afegint noves etiquetes mantenint la compatibilitat amb anteriors edicions del llenguatge.
- L'analitzador és un component estàndard: no és necessari crear un analitzador específic per a cada llenguatge. Això possibilita la utilització de qualsevol d'ells.
- És compatible: si qualsevol altre persona o entitat vol utilitzar el document creat en XML, és fàcil d'entendre la seva estructura i processar-lo, amb això es millora la compatibilitat entre aplicacions.

### 8.2.2 Definició de l'XML per a representar dades de GoogleMaps

Un cop APIS a cridat al middleware, aquest mitjançant el mètode *pintarMapa* definit a la classe *GoogleMaps* crida al SAXParser perquè reculli les dades de l'XML que defineix el mapa.

Per a crear aquest XML, s'ha hagut de pensar de quina manera es podria crear un llenguatge que cobris totes les necessitats i les dades que necessita l'API de GoogleMaps per a ser cridada sense cap tipus d'error. Alhora, aquesta estructura havia de ser compatible i similar a l'estructura de dades en Java on s'emmagatzemen les dades recollides, per tant el llenguatge seria molt semblant a una estructura de dades.

La nostra definició de l'XML per a representar les dades comença per definir en primer lloc les dades més importants e imprescindibles fins a arribar als detalls i informació addicional.

Per a definir un XML que representa mapes de GoogleMaps i identificar-lo com a tal, cal englobar tota la resta de tags dins d'aquests:

```
<GoogleMaps>  
  ---  
</GoogleMaps>
```



A partir d'això, el parser ja sap que les dades serviran per a ser incrustades en les funcions de l'API de GoogleMaps i així les podrà emmagatzemar en l'estructura de dades corresponent.

Seguint la jerarquia abans esmentada, per a que APIS pugui pintar un mapa en un ítem, primer de tot necessita conèixer el centre. Aquest centre consta d'una latitud, d'una longitud i un zoom. Aquesta estructura es representa així:

```
<Center>  
  <latitude> X.XXXXX </latitude>  
  <length>X.XXXXXX</length>  
  <zoom>X</zoom>  
</Center>
```

*Nota: Només hi haurà un centre per document.*

La latitud i la longitud són definits com valors decimals en coma flotant de les geocoordenades terrestres i al zoom li correspon un sencer que situa l'alçada a la qual s'ha de situar la càmera per sobre del nivell del mar. GoogleMaps defineix el zoom de 0 fins a 23.

Un cop ja tenim definit en quint punt del globus terrestre es mostrarà, el següent punt important és definir quina interacció es realitzarà a sobre del mapa, això es defineix per a resoldre possibles ambigüitats, ja que pot haver-hi els mateixos elements i informacions a sobre el mapa però el tipus de resposta i la manera d'interactuar amb aquests elements pot ser diferent.

El tipus d'interacció no requereix que estigui definida per l'especificació QTI, per tant podem afegir qualsevol interacció externa o personalitzada no contemplada.

El tipus d'interacció es defineix d'una manera còmoda i senzilla:

```
<InteractionType> Tipus de Interacció </InteractionType>
```

Els tipus d'interacció que reconeix el nostre parser són els següents. Més endavant els explicarem en detall:

- **LinesInPolygon:** Serveix per a definir un tipus d'interacció si un conjunt de línies estan dins d'un polígon.
- **OneClick:** Defineix una interacció per a detectar si un punt està dins d'una regió definida per un polígon.
- **Order:** Serveix per a definir una interacció d'ordenació d'elements.
- **Choice:** Serveix per a definir interaccions d'escollir entre diverses respostes.
- **DraggableMarkers:** Aquest tipus d'interacció dóna la possibilitat d'arrossegat una sèrie de marcadors sobre el mapa i deixar-los en una àrea determinada que li correspon unívocament.

Aquesta manera de definir les interaccions fa possible que en el moment que s'implementin més es puguin anar afegint a la llista.

Un cop s'ha definit on és el centre i quina és la manera (interacció) d'indicar la resposta dins del mapa, cal definir els elements de la interacció, és a dir, quina forma i quina informació tindran les respostes. Aquestes seran reconegudes perquè aniran entre aquests dos tags:

```
<Respostes></Respostes>
```

El primer tipus d'opció de resposta són els anomenats *OptionMarkers* o marcadors d'opció. Definir aquestes etiquetes tindrà com a conseqüència la creació d'uns marcadors del tipus GMarker.

La manera de definir-los és la següent:

```
<OptionMarker>  
  <image>ruta de l'arxiu</image>  
  <info> cadena de text </info>  
  <latitude> coma flotant </latitude>  
  <length> coma flotant </length>  
</OptionMarker>
```

La *image* serà la ruta d'un fitxer que conté una imatge i que en comptes de sortir un pivot vermell, sortirà una icona clickable, el tag *info* defineix el missatge que volem que surti quan l'usuari fa click en el marcador. Pot servir a mode de informació o a mode de feedback.

La *info* servirà també per a definir la resposta que processarà el middleware i que li enviarà a APIS, per tant cal que al fitxer XML on es crea *Item* hi aparegui la informació que surt en aquest tag com a resposta correcta. Per últim la *latitud* i la *longitud* és el punt on volem que surti el marcador en el mapa.

Aquest tipus de resposta de moment està contemplada per a resoldre interaccions de tipus *choice* i *order* encara que pot ser utilitzat per a posar informació i altres interaccions futures que es puguin implementar com per exemple arrossegant els marcadors a àrees determinades o a línies específiques.

Una altre tipus d'interacció en les respostes que es pot definir és la de dibuixar regions al mapa on es puguin fer alguna sèrie d'actuacions.

Per a definir una regió (polígon), es fa de la següent manera:

```
<Polygon> vèrtex del polígon </Polygon>
```

Un polígon està compost per una sèrie de punts (vèrtex) per tant cal definir aquests vèrtex.

L'estructura per a definir dins l'XML com es construeix un vèrtex és la següent:

```
<Vertex>  
  <ID>sender</ID>  
  <latitude>float</latitude>  
  <length>float</length>  
</Vertex>
```

Automàticament el middleware enllaçarà els punts entre si amb línies i construirà el polígon.

Per tant un punt a sobre el mapa necessita una latitud i una longitud a més d'un identificador del punt per motius de l'algorisme alhora de traduir aquest codi en JavaScript.

Cal anotar que dins d'un polígon es poden definir tants vèrtex com vulguem sempre que estiguin dins dels tags `<Polygon></Polygon>`

De la mateixa manera que es poden afegir polígons, també podem definir línies, l'estructura és similar

`<Line>` vèrtex de la línia `</Line>`

On el número de vèrtex és infinit i han de seguir la mateixa estructura abans esmentada.

Com veiem aquest sistema de dissenyar és escalable, ja que podem afegir més tipus de resposta a mida que es vagin dissenyant i implementant amb una correcta definició dels tags i modificació del parser.

En resum, aquesta estructura és orientada a objectes i així fem que sigui fàcilment compatible amb una estructura de classes en Java.

### 8.3 Estructura de dades del middleware

Un cop estan definides les dades dins l'arxiu XML, el middleware és l'encarregat d'interpretar aquest arxiu i emmagatzemar les dades obtingudes en les classes corresponents. Per tant aquí es veuen involucrats dos elements en el disseny de la implementació.

#### 8.3.1 SAXParser

El nostre middleware implementa un parser per a poder recollir les dades de l'XML que defineix el mapa. L'estructura d'aquest parser va detectant l'obertura i el tancament dels tags i a partir d'aquesta norma, va omplint les classes que li corresponen i necessita.

El SAXParser és cridat per la classe *GoogleMaps* des de el seu mètode *pintarMapa(arxiuXML)*, d'aquesta manera totes les dades que reculli el parser seran retornades a la classe que genera el codi en JavaScript necessari per a la generació del mapa.

L'algorisme del parser és el següent:

```
parser
{
    Si trobes un tag d'obertura
    {
        si el tag es igual a un dels coneguts i genera l'estructura
        {
            Estat = valor
        }
    }
    Si trobes un tag de tancar
    {
        Insereix les dades en l'estructura creada
    }
}
```

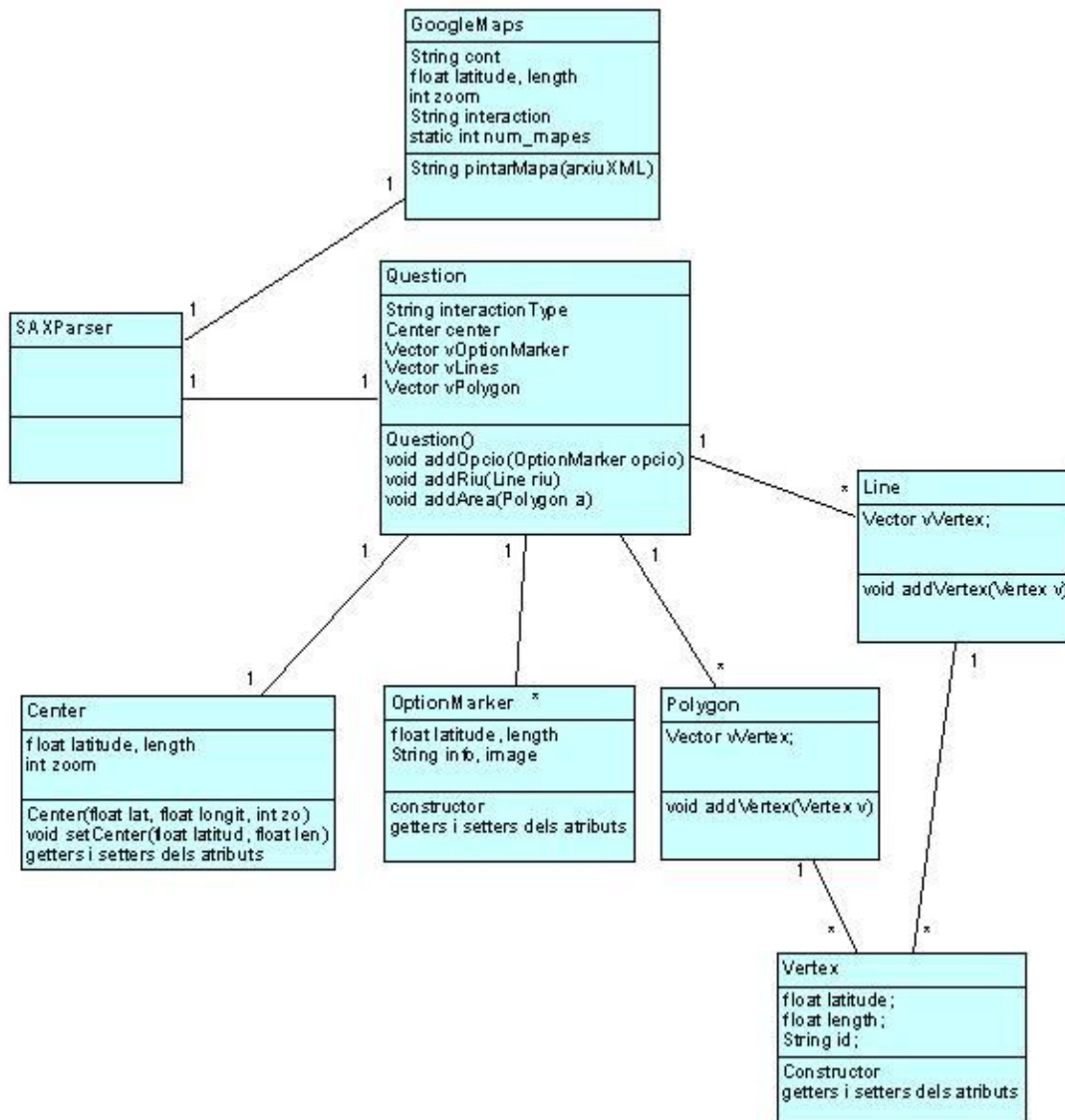
### 8.3.2 Estructura de classes

L'estructura de classes intenta assemblar-se el màxim possible a l'estructura de l'XML que defineix el mapa, d'aquesta manera fem que el codi sigui intuïtiu i de fàcil comprensió. L'objectiu és crear una classe per a cada un dels tags definits i després generar classes dels elements que contenen fent la relació entre els diferents elements mitjançant vectors.

Les classes que hi ha implementades a mode de resum són les següents:

- Center: conté una latitud, una longitud i un zoom.
- Line: compostat per un conjunt de vèrtex
- Polygon: compostat per un conjunt de vèrtex.
- OptionMarker: compostat per una icona, una posició, y una informació.
- SAXParser: És el parser en si, que ja hem explicat anteriorment.
- Question: conté tots els elements anteriorment anomenats.
- Vertex: Definit, per una latitud i una longitud.

I el diagrama que les relaciona és el següent:




Per tant sota la classe *GoogleMaps* que fa d'enllaç entre APIs i el middleware hi ha la descripció d'una pregunta amb interacció sobre el mapa.

Cada cop que es vol inserir un mapa dins d'un qüestionari, la classe *GoogleMaps* crea una instància de la classe *Question* (que és aquella que relaciona tots els elements necessaris d'un mapa), aquesta classe està composta per un centre (ja que el mapa només té un centre de situació, és a dir, on apunta la càmera), també guarda el tipus d'interacció en forma de cadena de caràcters i un vector per a emmagatzemar els marcadors d'opció, el conjunt de línies i el conjunt de polígons que volem superposar mitjançant els overlays a sobre del mapa respectivament. Per tant, es genera un vector per a cadascuna d'aquestes necessitats.

La classe *Center* està creada per tal d'emmagatzemar les coordenades geogràfiques exactes de qualsevol punt de la Terra, aquestes són:

- un *float* amb la latitud
- un *float* amb la longitud
- un *integer* amb el nivell de zoom que volem tenir.

La classe *OptionMarker* conté els següents elements:

- *float* latitud i longitud: serveix per a situar el marcador en un punt concret a sobre del mapa, cal notar que els marcadors no tenen necessitat de *zoom*, ja que la icona que mostren es veu igual de gran amb qualsevol nivell d'alçada, per tant si ens apropem la icona "taparà" menys terreny i si ens allunyem ens ocultarà més porció de terreny.
- *String* info: En aquesta cadena de caràcters, guardem la informació que conté el tag *info* de l'XML. Aquesta informació està pensada per a que la interfície mostri la resposta que ha seleccionat l'usuari i també per a enviar-li a APIS per a comparar amb la resposta correcta (més endavant s'explicarà com es fa la correcció de les preguntes).
- *String* image: Aquest atribut contindrà la ruta en forma de cadena de caràcters amb la icona personalitzada que es vol que es mostri. Per defecte GoogleMaps situa aquesta icona , però es pot posar qualsevol altra imatge en format GIF.

En quant a les classes *Line* i *Poligon* només direm que l'atribut que contenen és un Vector de la classe *Vertex* amb els seus mètodes per afegir a aquest el conjunt de vèrtex necessaris per a descriure la figura geomètrica.

Per últim, la classe *Vertex* està dissenyada així:

- *float* latitud i longitud: Com els marcadors d'opció, el vèrtex és un punt dins del mapa, per tant ha de contenir coordenades geogràfiques.
- *Int* ID: Per últim, per qüestions d'implementació amb el codi de GoogleMaps es necessita tenir identificat cada vèrtex del polígon o la línia, amb aquest identificador de tipus sencer.

Per acabar i a mode de resum veiem l'arquitectura completa i l'entorn del nostre sistema que dona el servei de GoogleMaps a APIS.

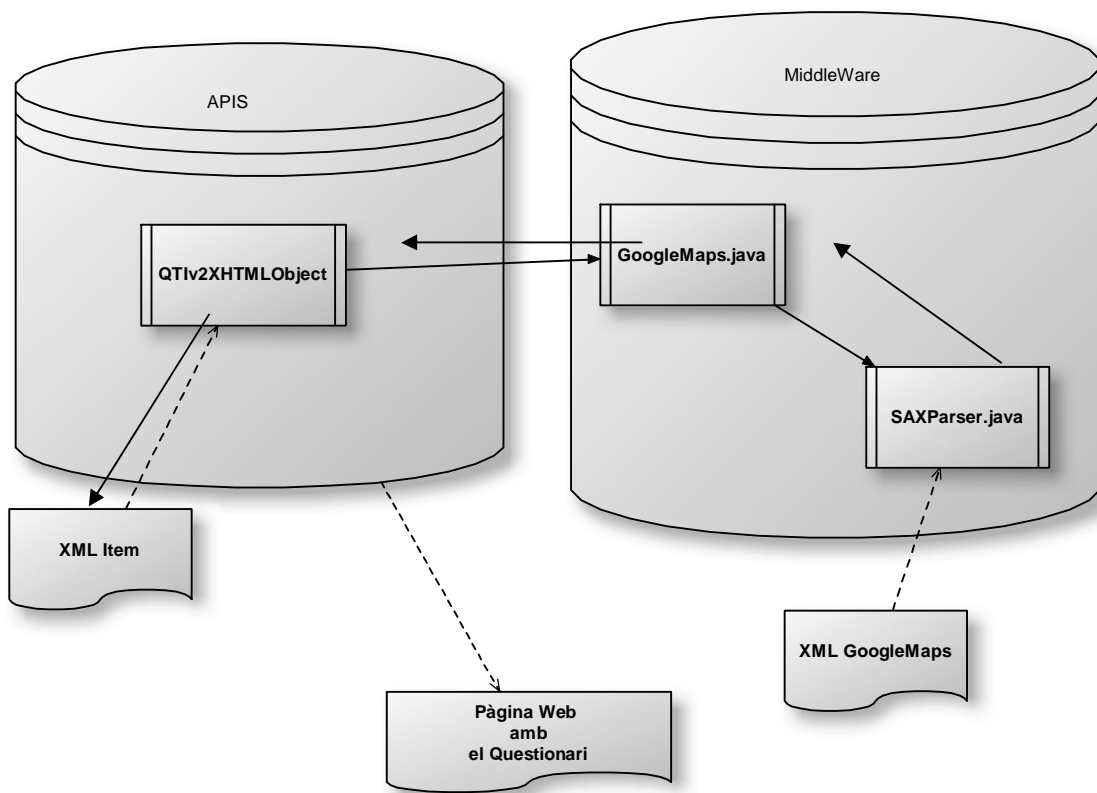


Fig 8.3.2.2 Entorn i relació d'APIS amb el MiddleWare





## 9. Les interaccions i processat de la resposta

---

En aquest capítol es farà una explicació de les interaccions que hi ha implementades i de com es realitza la correcció i processat de la resposta. Aquestes dues característiques estan directament relacionades, ja que l'alumne al interactuar i donar una resposta, el middleware també genera el codi amb el que APIS farà la correcció i avaluació d'aquesta.

Cal esmentar que en aquest capítol s'expliquen un conjunt d'interaccions a mode d'exemple (per a veure com funciona l'estructura i el disseny) però seguint les indicacions i el disseny que a continuació expliquem es poden implementar tantes interaccions com idees sorgeixin.

### 9.1 Interacció com a imatge

Aquesta interacció esdevé com a conseqüència i resultat de la connexió entre el middleware i APIS, per tant el servei de GoogleMaps s'utilitza només per a que faci la mateixa funcionalitat d'una imatge. Per tant la interacció amb la pregunta es fa a fora del mapa.

Encara que en un primer moment sembla que podria ser equivalent situar una imatge estàtica i un mapa de GoogleMaps a efectes docents, no és ben bé així, ja que GoogleMaps ens aporta una sèrie d'avantatges que una imatge estàtica no té.

En primer lloc a l'alumne se li presenta una imatge amb un nivell de detall que acostuma a ser molt superior al d'una representació cartogràfica, també l'alumne pot desplaçar-se per sobre el mapa explorant zones veïnes i fent zoom (com hem comentat abans des de nivell de carrer fins a vistes de mapa-mundi) per tal de situar-se millor en el context de la pregunta

Des del punt de vista del professor dóna la possibilitat de presentar-li a l'alumne un mapa físic, un mapa polític o un mapa amb els noms dels carrers per tal d'obrir en gran mesura el ventall de possibilitats a l'hora de fer preguntes.

La següent imatge mostra un exemple de com utilitzar el servei de GoogleMaps sense cap interacció de resposta a sobre el mapa:



Fig 9.1 Interacció choice fora del mapa

Per a definir aquesta pregunta, només cal construir un ítem amb la interacció que vulguem (en aquest cas un simpleChoice), situar en el tag imatge o en el tag map un arxiu XML definint les dades del mapa, i situar les possibles respostes dins de l'itemBody així:

```
<itemBody>
  <p>Look at the map and answer the question below; </p>
  <p>
    
  </p>
  <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
    <prompt>Which is the peninsula that you can see on the map:</prompt>
    <simpleChoice identifier="ChoiceA">Italian Peninsula</simpleChoice>
    <simpleChoice identifier="ChoiceB">Iberian Peninsula</simpleChoice>
    <simpleChoice identifier="ChoiceC">Balkan Peninsula </simpleChoice>
    <simpleChoice identifier="ChoiceD">Kola Peninsula </simpleChoice>
  </choiceInteraction>
</itemBody>
```

Com podem veure, tant la interacció com la correcció és responsabilitat d'APIS.

## 9.2 Interaccions dins del mapa

Quan APIS demana que es construeixi un mapa, a l'XML s'especifica quina interacció s'implementa; per tant el middleware quan li retorna el codi que fa que construeixi les dades, també li retorna el codi que haurà de generar per a la interacció i la correcció.

D'acord amb l'anterior, la responsabilitat d'aquestes dues accions (interacció i processat de la resposta) és traspasat per part d'APIS al middleware. Per a fer aquest traspàs de responsabilitats, cal definir en *l'item* a la zona on es defineixen les interaccions que el middleware serà qui construirà la interacció i processarà la resposta.

Aquest procés s'especifica de la següent manera:

```
<choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
  <prompt>Which is the largest city in Catalonia?</prompt>
  <simpleChoice identifier="GMapInteraction"></simpleChoice>
</choiceInteraction>
```

S'ha creat un identificador anomenat *GMapInteraction* on APIS, al detectar-lo, desarà les responsabilitats d'interacció al middleware. Una de les conseqüències que es desprèn és que en aquest moment ja no s'escriurà cap resposta a sota de la fotografia del mapa i per tant les respostes estaran definides dins d'aquest tal com ens havíem plantejat com a objectiu.

Tot seguit ho explicarem amb més nivell detall (veient algunes línies de codi per a major comprensió).

### 9.2.1 Interacció de Choice

La interacció de Choice intenta simular la mateixa interacció que hi ha definida en l'estàndard IMS QTI, és a dir, donat un enunciat hi ha un número definit de possibles respostes, de les quals una o diverses són les correctes. Per tant el fonament d'aquesta consisteix en mostrar un mapa amb unes característiques especificades i a sobre d'ell una sèrie de punts destacats que seran les possibles opcions de respostes que tindrà l'alumne per a resoldre l'enunciat i compondre la resposta.

D'aquesta manera, les possibles opcions alhora de respondre es veuran destacades a sobre del mapa mitjançant GMarkers (definites per una longitud i una latitud), els quals com havíem esmentat abans poden estar definits per la icona estàndard de GoogleMaps o per qualsevol imatge que el professor decideixi, si aquesta li aportarà la informació que vol mostrar a l'alumne, (una d'aquesta informació podria ser com s'anomena la resposta A,B,C,D...).

El GMarker també pot desplegar informació en ser clickat, així el professor pot mostrar a l'alumne qualsevol informació sobre la resposta que ha seleccionat.

En resum la interacció consisteix en presentar un mapa amb una sèrie d'opcions com a possibles respostes, l'alumne podrà fer click en totes les respostes i veure si hi ha alguna informació associada que li pugui ser útil. Un cop l'alumne hagi fet click sobre la resposta que vol validar haurà de prémer el botó "submit" per tal d'obtenir la correcció.

Veiem alguns exemples:

Look at the map and answer the question below;



Which is the largest city in Catalonia?

I a cada click podem veure informació relacionada:



Fig 9.2.1 Informació personalitzada a mostrar quan es fa click a sobre un GMarker

Quan l'alumne va clickant en el conjunt de possibilitats, cada cop que clicka en un marcadore, el middleware genera el codi necessari perquè APIS pugui recollir les dades, avaluar-les i mostrar la correcció pertinent.

Per tant el codi que hi ha per a que APIS pugui avaluar la resposta és el següent:

```
<input type='hidden' name='apisa_RESPONSE_0'  
      value='"+opt.getInfo()+"' >
```

Cal fer constatar que cada cop que l'alumne va clickant, es va generant el codi de cada resposta per a ser avaluat des de APIS; per aquest motiu, quan es prem

el botó *submit* agafarà les dades de l'últim click. Així afavorim que la manera de respondre per part de l'alumne sigui en aquest ordre: selecció de resposta correcta i validar la resposta.

El codi en JavaScript genera el codi HTML amb el valor de la resposta *value = opt.getInfo()* i compara aquest valor amb la resposta correcta que està definida a *l'item*. Si aquests dos Strings coincideixen donarà la resposta com a vàlida.

D'aquesta manera també es pot fer que sigui de resposta múltiple, únicament definint les respostes bones a *l'item* i generar el codi per tal de que reculli aquestes tres respostes en el moment de fer *submit*.

L'aplicació d'aquest tipus d'interaccions en el camp pedagògic de la geografia són diversos, la seva principal utilització derivarà en preguntes visuals tipus test, també poden ser utilitzats per enquestes.

### 9.2.2 Interacció d'Ordre

Funciona de manera molt similar a la interacció *choice* de múltiples respostes. A l'XML on es defineix la pregunta es definirà com una interacció d'ordre, també afegirà les respostes correctes en l'ordre adient. I a la interacció en si, es posarà l'atribut *GMapInteraction*.

De la banda de l'XML on es defineix el mapa, l'únic que cal fer és situar els marcadors (*GMarkers*) amb les possibles opcions a ordenar.

Aquesta interacció d'ordre es defineix de la següent manera:

Es mostra una pregunta la qual necessita resoldre's mitjançant l'ordenació d'una sèrie d'elements situats a sobre el mapa. L'usuari haurà de clicar en l'ordre correcte per tal de que la resposta pugui ser validada com a bona.

A mesura que vagi clickant a sobre dels marcadors (punts calents *GMarkers*) s'anirà mostrant l'ordre en el qual s'estan seleccionant. Un cop l'alumne decideixi que l'ordre que es mostra, és el que vol per a fer la seva resposta, premerà el botó *submit* per que APIS faci la correcció.

A mesura que es van seleccionant els marcadors, el middleware, com en l'anterior cas, anirà generant el codi HTML necessari per a que la pregunta pugui ser validada per APIS.

El codi on es fa això és el següent:

```
"b+=\"<input type='hidden' name='apisa_RESPONSE_'+num_click+'\"'  
      value='\"+opt.getInfo()+\"'>\"+opt.getInfo()+\"<BR>\"'  
      \";"+  
  
"document.getElementById(\"missatge\"+num_maps+\"").innerHTML  
      =\"'+b+'\"';\n"+
```

Cal fer notar que per a cada click a sobre del marcador, el middleware genera el codi de quin ha estat l'ordre en el qual s'ha clickat mitjançant la variable *num\_click*.

En conclusió la interfície per a la resolució d'aquests tipus de preguntes és molt similar a la donada per a resoldre preguntes de tipus *choice*.



Veiem un exemple:

Look at the map and answer the question below.



Barcelona  
Tarragona  
Lleida  
Girona

Order these Catalan cities according to their population

submit

Fig 9.2.2 Interacció d'ordre a sobre del mapa

De la mateixa manera podem anar implementant qualsevol altre interacció definida per IMS QTI v2.1 i APIS, sempre i quan es defineixi al ítem la/les resposta/es correcta/es, es defineixi la interacció com *GMapInteraction* i al middleware es generi el codi necessari per a poder fer la correcció per part d'APIS.

### 9.2.3 Interacció *PointIntoPolygon*

Fins ara l'alumne alhora d'interactuar amb el mapa, havia de seleccionar entre les respostes predeterminades. GoogleMaps però, dona una sèrie d'eines per tal de que qualsevol persona pugui interactuar lliurement amb el mapa. Una d'aquestes eines és la capacitat de dibuixar línies o detectar esdeveniments a sobre el mapa a més de poder-les processar.

D'aquesta manera, a partir d'aquesta secció, es volen introduir alguns exemples, els quals no coincideixen exactament amb una interacció clàssica definida per l'especificació.

La primera que hem desenvolupat s'anomena *PointIntoPolygon* en el qual a l'alumne en la pregunta se li mostra un mapa centrat en un punt, però sense cap tipus d'elements, punts calents, línies ni orientacions (únicament tindrà la orientació de l'enunciat).

Aquesta interacció es basa en que el professor defineix un polígon mitjançant l'XML i per altra banda l'alumne ha d'assenyalar aquest polígon. Per tant la

resposta es donarà per correcte si l'alumne clicka dins d'aquest polígon (que en el mapa es mostrarà invisible) i incorrecte si clicka fora del polígon.

Per tant el problema algorísmic d'aquesta interacció roman en detectar quan l'alumne està fent click a sobre de l'àrea seleccionada com a correcte i quan no.

Per a solucionar aquest problema vam analitzar l'entorn en el qual ens movem, que és un mapa mundial que està definit per coordenades geoespacionals, per tant dins d'aquest món es podrà aplicar qualsevol algorisme modelat en coordenades i en general qualsevol algorisme geomètric.

El problema geomètric al qual ens enfrontem és determinar si un punt (el click de l'alumne) està dins o fora d'un polígon definit. Aquest polígon pot ser còncau o convex.

El mètode aplicat per resoldre aquest problema és l'algorisme del "Punt en Polígon" basat en l'algorisme de la Corda de Jordan.

Més formalment l'algorisme consisteix en el següent:

Donat un punt  $P(x,y)$  i un polígon definit per una successió finita de coordenades cartesianes  $[v_1(x_1,y_1), v_2(x_2,y_2), \dots, v_n(x_n,y_n)]$ , el problema resideix en determinar si el punt pertany o no a l'interior del polígon.

En un principi, el procediment és senzill: és suficient traçant una semirecta auxiliar des del punt que volem analitzar fins a l'infinit en qualsevol direcció. Un cop l'hem traçat, només caldrà comptabilitzar el número d'interseccions d'aquesta semirecta amb els costats del polígon.

Si el resultat és un número parell o zero, el punt estarà a fora del polígon; pel contrari, si el còmput dona un número senar el punt  $P$  es trobarà dins del polígon.

Com a la figura següent podem veure:

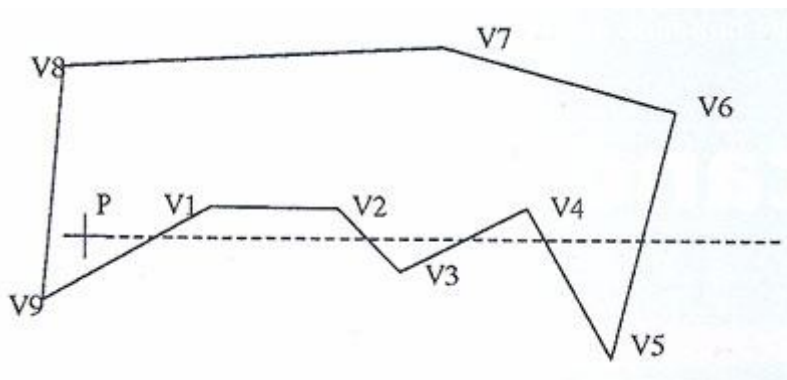


Fig 9.2.3.1 Com saber si un punt està dins o fora d'un polígon

Hem de veure que la lògica i el processat de la resposta, el fa el middleware implementat i per tant aquest li retorna a APIS la resposta processada i corregida, en aquest cas, APIS només dona l'avaluació.

La implementació d'aquest tipus d'interacció que no són específiques de l'especificació IMS QTI està modelada de la mateixa manera que una pregunta *simpleChoice* amb únicament dues possibles respostes: Si es fa click dins del

polígon és l'opció correcta i si es clicka fora del polígon es selecciona la resposta incorrecta.

A la següent imatge podem veure una interacció de tipus *PointIntoPolygon* on el polígon engloba la península itàlica amb un marge d'error:



Fig 9.2.3 Interacció punt en polígon

Un cop hem explicat que la gestió de la interacció i el processat de la resposta el fa el middleware, la correcció i avaluació, com sempre, el fa APIS. D'aquesta manera el nostre middleware genera el codi necessari per a que APIS pugui avaluar la resposta, donar la nota i les retroaccions possibles:

```
Detecció de l'event click dins del mapa
{
  si el poligon conté el punt
  {
    ..... <input type='hidden' name='apisa_RESPONSE_0' value='Correcte'>
  }
  sino
  {
    ..... <input type='hidden' name='apisa_RESPONSE_0' value='Incorrecte'>
  }
}
```

És útil des de el punt de vista pedagògic assenyalar el punt en el qual l'alumne està clickant en aquell moment, així té una visió més completa de la seva pròpia interacció amb el mapa.



Com que aquesta interacció no està específicament definida a IMS Question and Test Interoperability (QTI), les aplicacions des de el punt de vista pedagògic poden ser diversos:

- Reconeixement d'espais, monuments, àrees, etc.: Com que GoogleMaps proporciona mapes des de carrer fins a visions globals del món, podem fer que l'alumne identifiqui certs espais com carrers, edificis, penínsules, continents, muntanyes, ...
- Cerca d'espais, monuments, àrees, etc.: Degut a la llibertat de moviments que GoogleMaps aporta, es poden fer qüestionaris de cerca d'àrees, monuments, etc. Es pot donar una visió global del mapa, definir un temps màxim per a trobar cert lloc geogràfic o construcció i fer que mitjançant el zoom i movent-se pel mapa pugui trobar el que es demana a l'enunciat.
- Qualsevol tipus de pregunta en la qual s'hagi de assenyalar amb el ratolí una àrea determinada.

Per últim cal esmentar que la definició d'aquest polígon cal que es faci definint un marge d'error possible, ja que depenent del zoom, es pot veure una àrea molt gran i fàcilment clickable o una àrea molt petita on sortir-se del polígon pot ser molt fàcil. Per tant cal tenir en compte segons el zoom (si no és fix) la mida del polígon de resposta correcta.

#### **9.2.4 Interaccions dibuixant línies a sobre el mapa**

Una altra d'aquestes interaccions que poden ser molt útils pedagògicament parlant de cara a l'alumne i el professor, és la possibilitat de que l'alumne deixi la seva resposta pintada al mapa, és a dir, que pugui respondre traçant línies.

D'aquesta manera a l'alumne se li presentarà un enunciat i un mapa en el qual ha de respondre traçant una línia o un conjunt de línies per a delimitar, assenyalar o encerclar una resposta.

Per tant les utilitats d'aquest tipus d'interaccions són diversos, com per exemple:

- Assenyalar fronteres, delimitacions, separacions entre zones del mapa
- Encerclar, repassar monuments, zones del mapa, països,...
- Preguntar sobre rutes (per carretera o muntanya), tant a nivell de carrer com a nivell de mapa global.
- Relacionar unes zones amb unes altres.

La interacció es construeix mitjançant una àrea definida (que serà l'àrea correcta) que serà invisible per a l'alumne i una eina proporcionada per a poder pintar línies a sobre el mapa.

Per a comprovar si la resposta és correcta o no, l'algorisme comprova si aquestes línies (segments) formen part del polígon (estan dins completament) o fora per a determinar si la resposta és correcta o incorrecta respectivament.

Hi ha dues maneres de definir aquesta àrea on l'alumne obtindrà la resposta com a correcta:

- *Definir un polígon:* Amb aquest mètode, es defineix un polígon mitjançant una sèrie de punts (vèrtex) i per tant podem definir la figura que vulguem.
- *Definir un buffer:* D'aquesta manera es defineix una línia i una distància com a marge d'error i el middleware construeix un polígon seguint la forma de la línia marcada amb un marge d'error.

Veiem alguns exemples:

Look at the map and answer the question below



Find the shortest path between Arc del Triomf and France Station

submit

Fig 9.2.4.1 Definició de l'àrea correcta (en color blau)



Look at the map and answer the question below



Draw the frontier between Spain and Portugal

submit

Fig 9.2.4.2 Interacció situant línies dins àrees

Look at the map and answer the question below



Find the shortest path between Arc del Triomf and France Station

submit

Fig 9.2.4.3 Assenyalar rutes amb el mateix tipus d'interacció

### 9.2.5 Interacció arrossegant marcadors

Una altra de les possibilitats que ofereix GoogleMaps són els possibles esdeveniments que es poden detectar a sobre dels marcadors; aquests són:

- Click: Es llança un esdeveniment quan es fa *click* a sobre del marcador.
- Dblclick: Es llança aquest esdeveniment quan es fa doble *click* a sobre el marcador.
- Dragstart: Es llança l'esdeveniment quan es comença a arrossegar el marcador
- Drag: Es llança l'esdeveniment quan s'està arrossegant el marcador
- Dragend: Es llança l'esdeveniment quan es deixa anar el marcador

Per activar la possibilitat de que un marcador pugui ser arrossegat (*draggable*) cal activar-lo específicament d'aquesta manera en la funció corresponent:

```
GMarkerOptions.draggable = true.
```

Amb aquestes eines podem definir la interacció on a l'XML es definirà la interacció *DraggingElements* en el camp `<InteractionType>` i es col·locaran una sèrie de marcadors situats aleatòriament al mapa. Després mitjançant polígons s'especifiquen una sèrie de regions que marcaran una sèrie d'àrees a sobre del mapa transparents.

A cada regió transparent li correspondrà un marcador en concret, per tant la interacció amb l'alumne consisteix en situar dins de totes les àrees els seus respectius marcadors, tenint en compte que a cada àrea només li correspon un únic marcador.

Cal tenir en compte alhora de modelar l'arxiu XML que forma el mapa, definir els marcadors en el mateix ordre que es defineixen els polígons per que així després el middleware trobi la correspondència entre elements.

```
Marcador1....  
Marcador2....  
Marcador3....
```

```
Polygon 1...  
Polygon 2...  
Polygon 3...
```

L'alumne per tal de situar els marcadors dins de les regions haurà d'arrossegar-los a aquests des de la situació inicial fins a la final. Durant aquest trajecte s'han definit una sèrie de *listeners* per a donar certa retroacció i informació a l'alumne:



```

Per a cada marcador
{
    GEvent.addListener(marcador, "dragstart",function(){
        map.closeInfoWindow();
    });

    GEvent.addListener(marcador, "dragend", function (overlay,point){

        punt = marcador.getPoint();
        fem el test del punt amb el poligon;
        Si pertany el punt del marcador al poligon, llavors inserta el codi per a fer la correcció
        si no insterta el codi de que la resposta ha estat incorrecta
    });

    GEvent.addListener(marcador,"click",function(){
        marcador.mostraInformacióDelMarcador();
    });
}

```

Per tant, per a cada marcador detectem aquests tres esdeveniments i els seus *listeners* corresponents així quan l'alumne faci *click* a sobre del marcador, obtindrà informació d'on s'ha de posar aquell marcador (o qualsevol altre que cregui l'avaluador). Per altra banda quan deixi el marcador a un lloc determinat, automàticament es farà el test de si està ben situat aquell marcador en aquella regió.

Il·lustrarem aquesta interacció amb un exemple que s'ha implementat:

Look at the map and answer the question below;



Drag the elements into its correct place

submit

Fig 9.2.5 Marcadors per arrossegar i zones ressaltades on s'han de deixar

### 9.3 Complementant les interaccions (informació addicional)

GoogleMaps a sobre del mapa ofereix un tipus d'informació específica directament a sobre d'aquest segons el tipus de representació que escollim:

- Visió satèl·lit: Proveeix imatges i mapes físics de la Terra, tota la geografia física (en 2D, a Google Earth en 3D) que inclou rius, oceans, llacs, muntanyes, etc.
- Visió mapa: En aquest tipus d'informació amb *zooms* molt grans (propers al terra) es mostra informació dels carrers, carreteres, autopistes. Per altra banda també ofereix els topònims de pobles, ciutats, carreteres. A gran escala (amb *zooms molt petits*) la informació que ens dóna GoogleMaps és un mapa polític (països, regions, comunitats autònomes, departaments, etc.)

Per tant, tota aquesta informació sempre la tindrem disponible per a poder enfocar-la a l'ensenyament i l'avaluació. Hi ha de vegades però, que necessitarem informació addicional per tal de que la interacció sigui més comprensible, posar en context l'enunciat, orientar la resposta o simplement preguntar per coses que no són específiques del mapa.

Des d'aquest punt de vista, GoogleMaps ens ofereix la possibilitat d'afegir capes superposades al mapa (*Overlays*) els quals es situen utilitzant les coordenades geoespaciales i que mantenen la proporció segons l'escala a la qual l'estem veient, és a dir, canvien la seva dimensió segons des de quin zoom estem visualitzant el mapa. Aquest elements que ens proporciona GoogleMaps de tipus simple són els que ja hem vist amb anterioritat:

- Marcadors (*GMarkers*)
- Línies (*GPolyLines*)
- Polígons (*GPolygon*)

Cal destacar d'aquests tipus de capes simples, encara que mantenen les seves dimensions espaciales, podem fixar el seu gruix, és a dir, que si pintem una línia la veurem amb la mateixa amplada amb qualsevol valor de zoom.

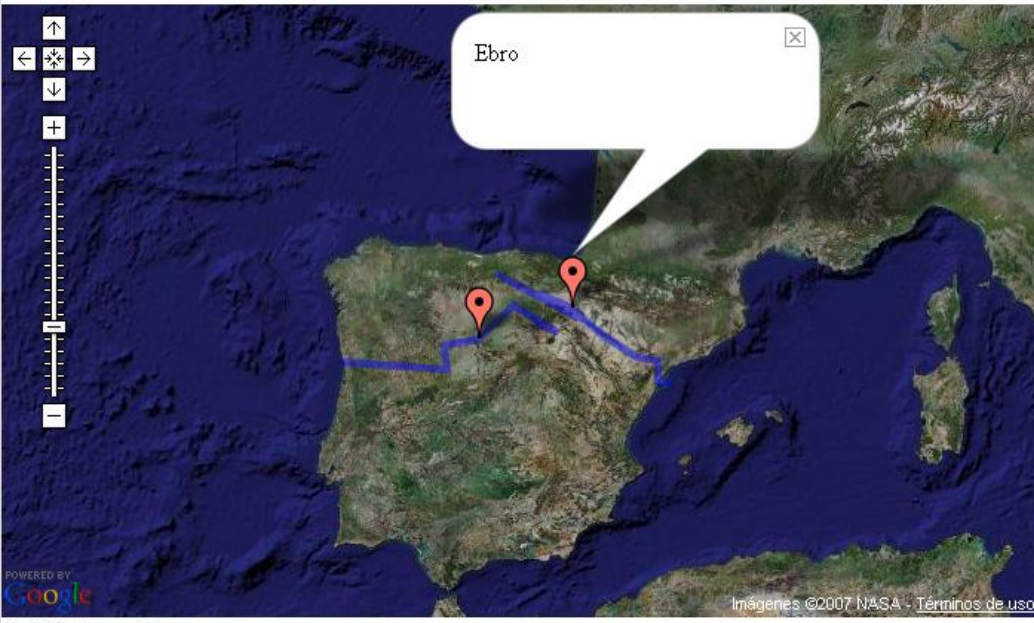
D'aquesta manera podrem destacar elements que volem que siguin visibles des de qualsevol distància de la Terra, definir un mapa amb informació personalitzada, no fer visible qualsevol element del mapa mitjançant un polígon opac, etc.

Com hem vist en anteriors apartats, aquestes capes simples han estat utilitzades per a definir interaccions o fer operacions lògiques amb elles per tal de determinar les respostes seleccionades per l'alumne, ara només s'utilitzaran amb l'objectiu de donar informació útil a l'alumne.



Aquesta utilitat ens pot servir en alguns casos per a fer preguntes de l'estil:

Look at the map and answer the question below;



Find the longest river

submit

Fig 9.3.1 Els overlays serveixen per a destacar rius i fer preguntes sobre ells.

O també:

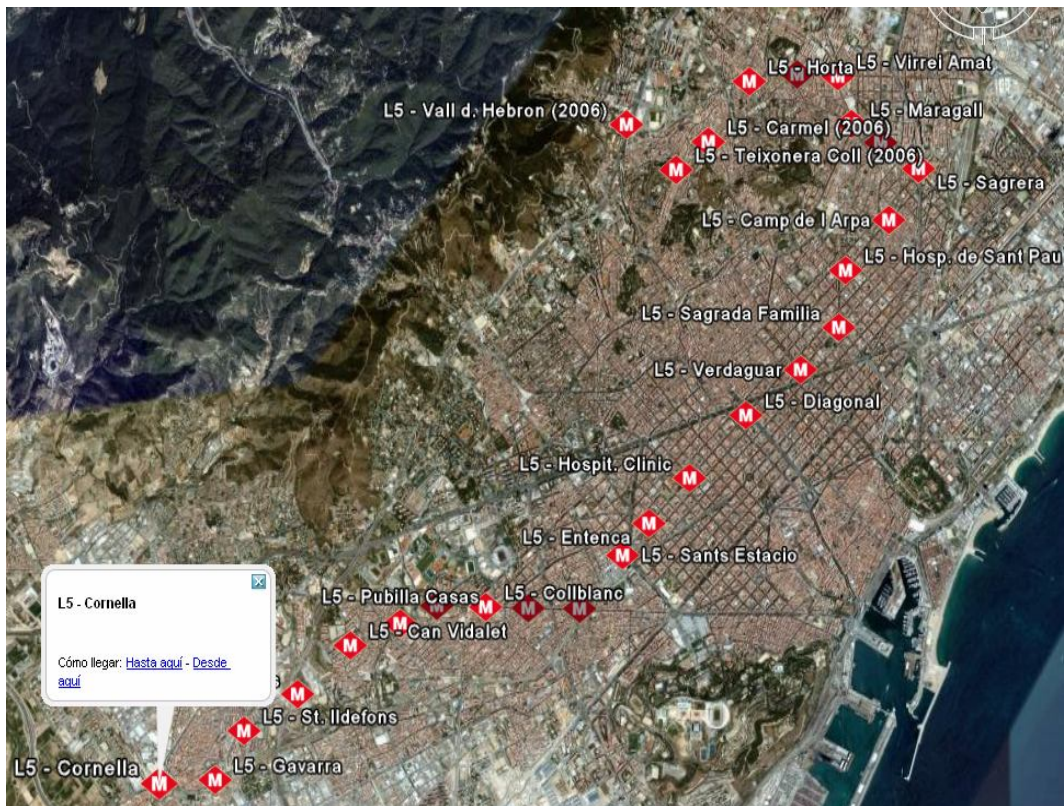


Fig 9.3.2 Mapa del metro de Barcelona (Línia 5)

A l'exemple anterior hem vist una capa on es mostren les parades del metro de Barcelona, aquest tipus de capes es poden definir directament amb l'XML que defineix les característiques del mapa com hem explicat abans, mitjançant marcadors i icones, o també hi ha una altra manera d'incorporar-los mitjançant arxius *kml*.

### 9.3.1 Els arxius KML

Keyhole Markup Language és un llenguatge de marcat basat en XML per a representar dades geogràfiques en tres dimensions. Està desenvolupat per Google amb la finalitat de ser utilitzat originàriament en Google Earth encara que actualment també pot ser interpretat per GoogleMaps.

La seva gramàtica conté moltes similituds amb la gramàtica de GML (*Geography Markup Language*). Els fitxers KML al ser distribuïts són empaquetats i comprimits com a fitxers KMZ.

Un fitxer KML especifica una característica o un conjunt d'aquestes (llocs, imatges, polígons, línies, etc.) per al mapa. Conté un títol, una descripció bàsica del lloc, les seves coordenades (latitud i longitud) i qualsevol altra informació addicional.

GoogleMaps pot actualment (22/05/2007) llegir fitxers KML originàriament definits per a Google Earth. Això significa que les dades creades i compartides per usuaris de Google Earth poden ser visualitzades també a GoogleMaps.

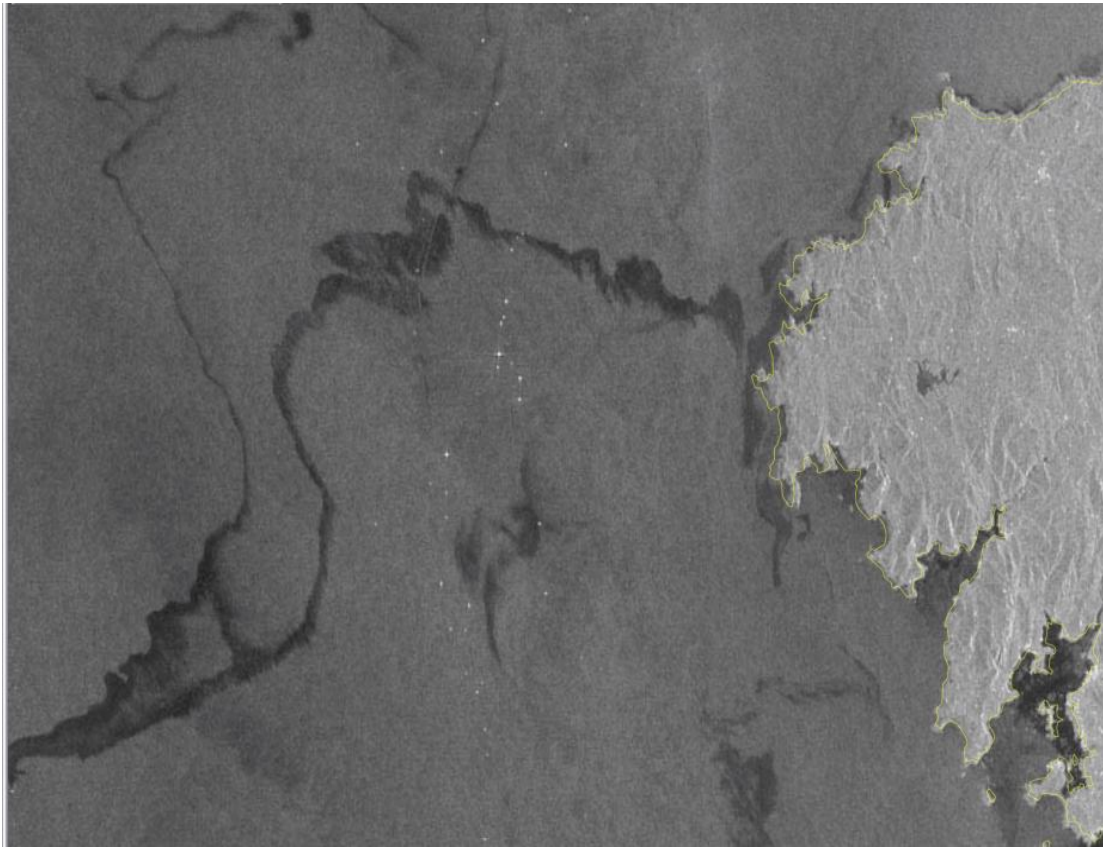
Encara que només suporta KML definits amb línies, punts, polígons i icones encara que properament es podran afegir KMLs amb capes superposades de fons, carpetes i visibilitats que faran possible superposar capes com les que a continuació es mostren i que seran una eina molt potent pedagògicament parlant:



Fig 9.3.3 Mapa actualitzat (cada 5 minuts) del trànsit de Barcelona, encara només per a GoogleEarth



O també:



*Fig 9.3.4 Abast de l'accident del Prestige a les costes de Galícia (Taques d'oli)*

Amb això també augmentarà la capacitat de superposar informació i també mitjançant webs per a compartir els materials KML com [www.gearthacks.com](http://www.gearthacks.com) es podrà accedir a major nombre de capes amb informació personalitzada.



## 10. Conclusions i treball futur

---

### 10.1 Conclusions

La importància dels mapes en l'estudi de la Geografia esdevé de capital importància i l'àmbit de l'e-Learning no ha d'estar al marge d'aquesta situació.

Actualment determinades empreses i organitzacions aprofitant les noves tecnologies ens aporten unes eines molt útils que creiem necessari que s'incorporin com a elements aprofitables de cara a l'estudi de determinades àrees i especialitats; en aquest cas del desenvolupament d'especialitats com la Geografia, la situació espacial, orientació, coneixement del medi que ens envolta, etc.

En concret una de les primeres fases del projecte era l'estudi de l'entorn de l'e-Learning i quines possibilitats dóna actualment per a la creació de preguntes i qüestionaris, quins tipus d'interaccions hi ha disponibles i quines hi haurà contemplades en futures versions de l'especificació. En aquest entorn vàrem veure que hi havia un espai per a incorporar nous serveis externs a la pròpia especificació.

Un d'aquests serveis és GoogleMaps que ens dóna un servei de imatges del planeta prou precís com per a poder desenvolupar preguntes complexes a sobre d'ell. A més d'això GoogleMaps ens proporciona per si mateix certa informació política, de carrers i geografia física que pot ser útil en determinades circumstàncies alhora d'elaborar qüestionaris i preguntes.

De la mateixa manera també creiem que el servei ens aporta una eina molt potent de cara a donar facilitats als docents alhora d'accedir a les imatges del planeta, ja que amb un sol moviment de ratolí pots arrossegar el mapa i desplaçar-te per tota la Terra per a obtenir qualsevol fotografia aèria i a qualsevol alçada (des de nivell de carrer fins a un mapa mundial), per tant, com ja hem comentat, el professor, alhora de preparar l'assignatura, no tindrà que carregar un conjunt de fotografies estàtiques i amb informació específica, només haurà de cridar al servei, accedir al punt de la Terra que vulgui, fixar la camera a l'alçada desitjada, seleccionar el tipus d'interacció i l'enunciat de la pregunta.

GoogleMaps dóna la possibilitat de personalitzar els mapes amb capes superposades, d'aquesta manera es proporciona una gran flexibilitat al professor per afegir dades que cregui importants a sobre d'aquest.

A més al ser una API pública hi ha una comunitat molt estesa creant funcionalitats, capes i altres capacitats que poden ser incorporades per part del professor simplement afegint un arxiu KML. Per aquest motiu creiem que la solució proposada amb la realització d'aquest projecte és molt útil i avantatjosa, facilitant així la tasca del professor alhora de recuperar i seleccionar material per als qüestionaris.

Des del punt de vista pedagògic i de l'alumne creiem que aquest projecte pot facilitar i fer més eficient l'aprenentatge de matèries com la Geografia.

Un primer avantatge de la incorporació d'aquest servei dins d'un player que interpreta l'especificació Question & Test Interoperability com per exemple APIS, és que les imatges al ser fotografies aèries realitzades des de satèl·lits donen un realisme i espectacularitat que una fotografia estàtica no pot donar. Així aquesta

espectacularitat incideix directament l'atenció de l'alumne, el que comporta un augment de la seva motivació. En conseqüència l'eficiència pedagògica és més important.

Un altre factor és el realisme que aporten, també incideix favorablement en l'atenció i motivació de l'alumne i a més dona la possibilitat que l'alumne pugui buscar punts de referència explorant i desplaçant el mapa i així poder memoritzar i aprendre millor el concepte al qual s'està interpellant. En resum s'enfronta directament amb imatges reals a escala i no amb reconstruccions figurades.

D'aquí es desprèn la gran interactivitat entre la pregunta i l'alumne, ja que en una fotografia normal no es pot fer zoom, no es pot escollir el nivell de detall amb el qual el vols veure o el tipus de mapa. Això afavorirà l'aprenentatge i les finalitats pedagògiques.

En conclusió, afegint GoogleMaps a APIS s'aconsegueixen mapes amb unes possibilitats d'interaccions molt potents i molt intuïtives que creiem que són millors i més versàtils de les que ens dona una imatge estàtica, ja que es treballa directament sobre "el món real" degut al realisme que proporcionen les fotografies del servei de Google.

De la part tècnica creiem que s'ha aconseguit un prototip de middleware el suficientment abstracte i aïllat per tal de no haver de modificar en gaire mesura la seva estructura fonamental, de fet no s'han creat noves classes dins d'APIS i només s'ha modificat una funció que crida al middleware, de la mateixa manera la resposta la genera aquest que li envia a APIS, i l'aplicació és qui la processa i dona la nota seguint el format preestablert.

Aquest disseny (incorporar un middleware) ens aporta també la possibilitat d'afegir altres serveis que puguin ser útils en l'àmbit pedagògic de manera modular i independent. Així podem fer que aplicacions que en un primer moment no són dissenyades per a l'educació, puguin treballar en aquest àmbit complementant-se amb APIS.

Les interaccions que s'han implementat a tall d'exemple mostren noves maneres de contestar per part de l'alumne seguint la metodologia clàssica d'interaccions. Els exemples que hem proposat donen la possibilitat de clicar sobre marcadors que determinen les possibles respostes o ordenar una sèrie de marcadors. Aquests exemples intenten imitar, donant una perspectiva més ambiciosa, les interaccions contemplades a QTI.

La implementació d'uns altres tipus de preguntes com:

- clicar lliurement sobre el mapa cercant un determinat emplaçament,
- determinar mitjançant línies fronteres entre països,
- determinar rutes entre carrers,
- arrossegar i situar uns marcadors a sobre de regions delimitades transparents,
- afegir capes (*overlays*) i fitxers KML amb informació addicional a sobre el mapa

ens obren noves vies d'interaccions més naturals amb la informació geogràfica.

## 10.2 Treball futur

Com hem esmentat anteriorment, la idea d'incorporar un middleware dins APIS possibilita la interconnexió d'aquest player amb altres serveis externs, en conseqüència, un treball futur seria connectar altres serveis externs mitjançant una API pública o qualsevol programari de codi lliure (com per exemple l'equivalent a GoogleMaps en open-source com OpenStreetMap proporcionat per MapServer [33]) i obert que pugui tenir una aplicació didàctica i educativa en qualsevol altre camp seguint aquesta mateixa filosofia.

En segon lloc, encara que hem desenvolupat una sèrie d'interaccions clàssiques i unes altres adaptades a GoogleMaps amb les eines que ofereix, també es podrien estudiar, dissenyar i implementar altres tipus d'interaccions per aprofitar més el potencial d'aquest servei, per això caldrà estar atent a futures versions de l'API i les noves possibilitats que pot oferir.

També caldria acabar de definir bé la sintaxi dels fitxers XML que contenen la informació del mapa, per així a banda de dir quin tipus d'interacció hi ha, on es situa el mapa i quins elements per interactuar hi ha a sobre d'ells, també el professor hauria de poder definir per exemple, quin tipus de mapa vol mostrar en cada moment (o si l'alumne pot escollir mitjançant uns botons entre *street map*, *satellite map* o *hybrid map*), si es pot visitar tot el mapa mundial, definir uns límits màxims i mínims de zoom, veure si es pot augmentar el zoom amb la roda del ratolí per a donar més llibertat a l'alumne per explorar el mapa. Totes aquestes qüestions de fàcil modificació donen un caire més concret que pot delimitar més quin tipus de pregunta es pot fer i per a quin objectiu es vol utilitzar el tros de mapa en la pregunta.

Enllaçant-lo amb l'anterior punt exposat, un altre treball futur que seria molt útil seria el disseny i implementació d'un editor, ja que alhora de definir on es centraran els mapes i on seran els elements cal especificar exactament quina és la longitud i latitud de l'element i per tant el professor hauria d'extreure aquestes dades primer i després plasmar-les a l'XML, amb un editor, el professor intuïtivament arrossegaria el mapa en la part que vol mostrar, amb una barra o amb la roda del ratolí definiria el zoom i amb un menú aniria seleccionant els elements a mostrar i així es generaria automàticament l'XML amb les dades, sense que el docent s'hagi d'implicar amb latituds, graus i decimals.



## Bibliografía

---

### L'eLearning

[1] David Griffiths, Josep Blat, Rocío García y Sergio Sayago. La aportación de IMS Learning Design a la creación de recursos pedagógicos reutilizables. Universitat Pompeu Fabra. [http://spdece.uah.es/papers/Griffiths\\_Final.pdf](http://spdece.uah.es/papers/Griffiths_Final.pdf)

[2] <http://www.elearningworkshops.com/>

[3] <http://es.wikipedia.org/wiki/E-learning>

[4] <http://www.imslobal.org/>

[5] <http://www.imslobal.org/pressreleases/pr070212.html>

[6] <http://ares.cnice.mec.es/informes/11/contenido/25.htm>

### Question and Test Interoperability

[7] <http://www.imslobal.org/question/>

[8]

[http://gauss.topografia.upm.es/~m.manso/docs/Estandar\\_QTI\\_Descripcion.pdf](http://gauss.topografia.upm.es/~m.manso/docs/Estandar_QTI_Descripcion.pdf)

[9] <http://ares.cnice.mec.es/informes/16/contenido/31.html>

[10]

<http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-36/briefing-36-A5.doc> -

[11] [http://edutechwiki.unige.ch/en/IMS\\_Question\\_and\\_Test\\_Interoperability](http://edutechwiki.unige.ch/en/IMS_Question_and_Test_Interoperability)

### GoogleMaps

[12] [http://es.wikipedia.org/wiki/Google\\_Maps](http://es.wikipedia.org/wiki/Google_Maps)

[13] <http://www.google.com/apis/maps/>

[14] <http://www.google.com/apis/maps/documentation/reference.html>

[15] <http://groups.google.com/group/Google-Maps-API>

[16] <http://googlemapsapi.blogspot.com/>

[17] <http://www.wingo.com/gmaps/examples.html>

[18] <http://koti.mbnet.fi/ojalesa/exam/index.html>

[19] <http://www.econym.demon.co.uk/googlemaps/>

### Javascript

[21] <http://login.osirislms.com/offline/javascript/index.htm>

[22]

<http://www.lawebdelprogramador.com/cursos/mostrar.php?id=45&texto=JavaScript>

### Java

[23]

<http://www.lawebdelprogramador.com/cursos/mostrar.php?id=44&texto=Java>

[24] Curso de Programación en Java. Francisco Javier Ceballos. Ed. Ra-MA. 2004

## Geografia

- [25] <http://www.ceu.unam.mx/ponsemloc/ponencias/29.html>
- [26] [http://es.wikipedia.org/wiki/Coordenadas\\_geogr%C3%A1ficas](http://es.wikipedia.org/wiki/Coordenadas_geogr%C3%A1ficas)

## APIS

- [27] <http://www.jisc.ac.uk/whatwedo/projects/apis.aspx>
- [28] Sistema per a la gestió de qüestionaris QTI en el context d'una arquitectura orientada a serveis. Helena Batlle. Projecte Final de Carrera. Universitat Pompeu Fabra. 2006

## XML

- [29] <http://es.wikipedia.org/wiki/XML>
- [30] <http://www.dat.etsit.upm.es/~abarbero/curso/xml/xmltutorial.html>

## ALTRES

- [31] [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=611](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=611)
- [32] <http://mapserver.gis.umn.edu/>
- [33] International Organization for Standardization: *Geographic information - Web Map server interface standard (ISO 19128:2005)*. 2005.

**NOTA:** Totes les webs han estat consultades per últim cop al Juny del 2007



## **ANNEXES**



## Glossari

---

- ADL:** *Advanced Distributed Learning*. <http://www.adlnet.org/>. Al 1997 el Departament de Defensa dels Estats Units i l'Oficina de Ciència i Polítiques Tecnològiques de la Casa Blanca llancen la iniciativa (ADL). La missió d' ADL és proveir accés de la més alta qualitat en educació i entrenament, en qualsevol lloc i en qualsevol moment. Per a complir amb aquests objectius creen el model SCORM.
- AICC:** ( *Aviation Industry Computer-Based Training Comitee* ). Es reconeix com una dels precursors de l'estandardització de materials de l'entrenament professional.
- ARIADNE:** ( *Alliance of Remote Instructional Authoring and Distribution Networks for Europe* ), <http://www.ariadne-eu.org>. És un projecte d'investigació i de desenvolupament tecnològic de telemàtica per a l'educació i l'entrenament, patrocinat per la Unió Europea. El projecte s'enfoca al desenvolupament d'eines i metodologies per a produir, administrar i reutilitzar elements pedagògics basats en computadora, així com la funció d'entrenament a distància.
- CEN:** (*Comité Européen de Normalisation*). És una organització sense ànim de lucre amb l'objectiu de fomentar el benestar dels ciutadans europeus i el medi ambient proporcionant estàndards i especificacions per al desenvolupament i distribució.
- GML:** (*Llenguatge de Marcat Geogràfic*). Subllenguatge d'XML descrit amb una gramàtica XML\_Schema per al modelatge, transport i emmagatzematge d'informació geogràfica.
- Estàndards de jure:** Un estàndard de facto és aquell patró o norma que es caracteritza per no haver estat legitimat per un organisme internacional d'estandardització. Pel contrari es tracta d'una norma normalment acceptada i àmpliament utilitzada per iniciativa pròpia d'un gran nombre d'interessats. Acostumen a convertir-se en conseqüència estàndards de facto.
- Estàndards de facto:** Són generats per un comitè amb *estatus* legal i tenen el recolzament de governs o institucions per a produir estàndards.
- IEEE/LTSC** ( *Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee* ), <http://www.ieee.org>. L'IEEE és una associació internacional, la missió de la qual és promoure els processos d'enginyeria per a la creació, desenvolupament, integració, compartició i aplicació del coneixement sobre tecnologies electròniques i d'informació. Dintre de la seva organització hi ha el Comitè d'Estàndards per a Tecnologia de l'Aprenentatge o LTSC, que s'encarrega de desenvolupar estàndards tècnics, recomanacions i guies per a la tecnologia educativa.
- ISO:** (*International Organization for Standarization*). És una organització no governamental composta per representants d'organismes de

normalització nacionals que produeix normes internacionals industrials i comercials.

- Metallenguatge: Llenguatge utilitzat per a fer referència a altres llenguatges.
- Namespace: Conjunt de noms els quals tots són noms únics.
- Repositoris: Lloc centralitzat on s'emmagatzema i manté informació digital, normalment bases de dades o arxius informàtics.
- Sexagesimal: El grau sexagesimal, com unitat de sistema de mida d'angles, està definit partint de la base que un angle recte té  $90^\circ$  sexagesimals i els seus divisors, el minut sexagesimal i el segon sexagesimal estan definits de la següent manera:

- $1 \text{ angle recte} = 90^\circ$  (graus sexagesimals).
- $1 \text{ grau sexagesimal} = 60'$  (minuts sexagesimals).
- $1 \text{ minut sexagesimal} = 60''$  (segons sexagesimals).

De la mateixa manera, una quantitat de graus es pot expressar de manera decimal (GoogleMaps ho fa així), separant la part sencera de la decimal de la següent manera:

- $23,2345^\circ$
- $12,32^\circ$
- $-50,265^\circ$
- $123,696^\circ$

- W3C: (*World Wide Web Consortium*). Consorci per a la producció estàndards per a la WWW. Els més destacats són: URL, HTTP i HTML.

---

### Enhancing IMS QTI assessment with web maps

---

José Bouzo, Helena Batlle, Toni Navarrete, Josep Blat

Departament of Information and Communication Technologies, Universitat Pompeu Fabra

Passeig de Circumval·lació, 8. 08003 Barcelona, Spain

E-mail: jbouzo4@gmail.com, helena.batlle@upf.edu, toni.navarrete@upf.edu,  
josep.blat@upf.edu

**Abstract:** Although the importance of maps in Geography education, current eLearning systems do not take benefit from the possibilities that web maps offer. In this paper we describe an IMS QTI assessment engine enhanced with web maps from Google Maps. The system enables the user to interact with the map to answer questions, providing a more natural interface for geographic information. The concept of map interaction has been introduced to represent the different ways of processing the student actions on the map. Depending on the selected map interaction, different spatial operations are applied to validate the correctness of responses.

**Keywords:** eLearning, assessment, IMS QTI, Geographic Information, Google Maps, Web Map Service

---

#### 1 Introduction

Humans have used maps to represent parts of the Earth since the Stone Age. During these millennia, the science of Cartography has evolved from portraying simple representations on walls to the extremely precise real-time-populated interactive web maps we can find today. Maps are the most widely used tool in Geography to represent those features and phenomena that have a spatial component. Maps are also used in other disciplines where space may be an influential factor such as History, Economics and Business, Sociology, Politics, Biology or Environmental Science, among others.

From a pedagogical perspective, maps also play a key role in Geography education. In fact, understanding maps is a key competence that children have to acquire. Maps are widely used in Geography learning activities and contents. Maps are needed by K-12 education students to understand their local and global environment, and how human activities take place in these environments. Higher education and long-life learning also make a frequent use of maps.

However, the use of maps in eLearning has been very reduced, mainly constrained to non-interactive still images. Web map servers offer the possibility of incorporating more elaborated maps in the learning process, with more interactivity and learning possibilities.

The importance of maps in Geography education can be extended to assessment activities, since maps are usually needed to evaluate the acquisition of competences in Geography and related fields. In this work we focus on how maps can be used in the process of assessment in an eLearning platform. We are interested in an assessment system supporting the visualization of web maps and providing the typical interaction tools such as zoom in or zoom out. Furthermore, the system has to enable students to answer questions by interacting with the map in different ways:

- clicking on key elements on the map
- sketching points, lines or polygons on the map to represent geographic elements, such as cities, rivers or countries respectively.

Section 2 describes the main specification for interoperable assessment, IMS QTI. It also describes APIS, the engine that we have upgraded and used in our experiments with mapping services for assessment. Section 3 focuses on the main approaches for serving maps on the web: the OGC Web Map Service specification and Google Maps. Section 4 presents our approach of combining the APIS engine and Google Maps through a middleware that we have developed. Finally, Section 5 summarizes the main conclusions and states some lines for future work.

## 2. IMS QTI and the APIS engine

Question and Test Interoperability (QTI) [1] is the IMS specification for assessment. It provides a data model for the representation of questions (items) and tests and their corresponding outcomes. It enables the exchange of questions and tests between authoring tools, item banks, test composition tools, learning systems or assessment delivery systems, to name a few. Although the model is defined in abstract terms, an XML implementation is also provided in the specification.

The main elements of the QTI data model are:

- **Item:** it is the smallest interchangeable QTI element that stores the question presented to the user along with the associated metadata such as the reproduction instructions, user answers processing mode, hints, and feedback.
- **Section:** it represents a composite part of the assessment test or exam.
- **Test:** it is an entire QTI instance that embodies a single assessment test. Its structure is divided into sections and subsections and contains sequential information along with the method(s) to use for combining individual questions scores/marks to form the overall test grade.

The last two versions of the specification are 2.0 and 2.1. QTI 2.0 focuses on the representation of individual questions, introducing a long list of interactions. An interaction describes how the user “interacts” with an individual question, and can be seen as a type of question. An example of interaction, probably the most usual, is the simple choice question, where the user can select just one from several possible responses to the question, and where just one of them is correct. On the other hand, QTI 2.1 (still in public draft phase) deals with tests and their internal organization in sections. QTI 2.1 also defines complex ways of producing results reports for a whole test.

An engine is necessary to run QTI tests. A QTI engine is the software component responsible to manage the QTI data model, processing the XML file and generating the outcomes according to the user actions. There are two main open source implementations of QTI engines that we describe now.

The APIS (Assessment Provision through Interoperable Segments) QTI 2.0 engine [2] was originally created by Strathclyde University. A modular item-rendering engine was defined, although only some of the most widely used interactions were implemented. This engine addresses the operations required by potential tools defined in the Open Knowledge Initiative (OKI) [3] and IMS Web Services [4].

The R2Q2 (Rendering and Response processing services for QTIv2 questions) project [5,6] has been developed more recently by the University of Southampton. It is a new implementation built from scratch, aiming at providing a complete renderer and response processing engine, properly structured. Due to its function-modular design (Renderer, Processor,...) and use of internal Web Services, the system facilitates future enhancement and can be changed to suit any application.

However, both implementations are limited to the QTI 2.0 specification, and consequently can only process individual questions. We have largely upgraded the APIS engine to making it compliant with QTI 2.1, introducing new functionalities mainly referring to: test context instead of just questions, a wide range of new elements related to the test level, new and more complex response processing and new types of interactions. This new version of APIS is available at Source Forge at <http://sourceforge.net/projects/newapis>. A more detailed description of the structure of the new APIS engine can be found in [7].

This upgraded APIS engine will be the base for processing those questions needing mapping services, as it will be described in Section 4.

## 3. Google Maps and other mapping services

Serving maps on the Internet has become very popular in the last few years. A multitude of implementations of mapping servers exists, both commercial and open source, using different programming languages. In this context of heterogeneous tools, the Open Geospatial Consortium (OGC) is the international organization, integrated by companies, governmental agencies and universities, that is leading the definition of consensus standards in the field of geospatial services. Several OGC specifications have become ISO standards. In the domain of map servers, OGC has developed the Web Map Service specification (WMS) [8], which defines a service-based interface for a standard map server consisting of the following three main services:

- *GetCapabilities*, which returns metadata related to the server
- *GetMap*, which returns a map (an image) given some parameters as the coordinates of the centre of the map, zoom level, ...
- *GetFeatureInfo*, which returns the information of a given feature(s) appearing on the map, given a pair of pixel coordinates

Any WMS-compliant server may define other services, but have to implement these three basic ones, that in fact are sufficient to develop an interactive on-line map. There are several commercial and open source implementations of WMS. MapServer [9], developed by the University of Minnesota, is the most widely used open source implementation of WMS. WMS has been adopted by ISO as the ISO 19128 Geographic information - Web Map server interface [10], becoming a normative international standard.

However, the appearance of Google Earth and Google Maps has change the world of map servers. Google Maps is a simple web-based map viewer, while Google Earth is a richer desktop application that provides an attractive 3D visualization, as well as other extra features. Both tools offer a free world-wide cartography including street-level information for most Western countries, as well as satellite image at different resolutions. Google has also defined KML, a simple XML-based file format for storing vector-based (points, lines and polygons) geographic information. Users around the world have produced KML files to show their favourite spots, routes, etc, that can be seen on the free cartography that Google offers.

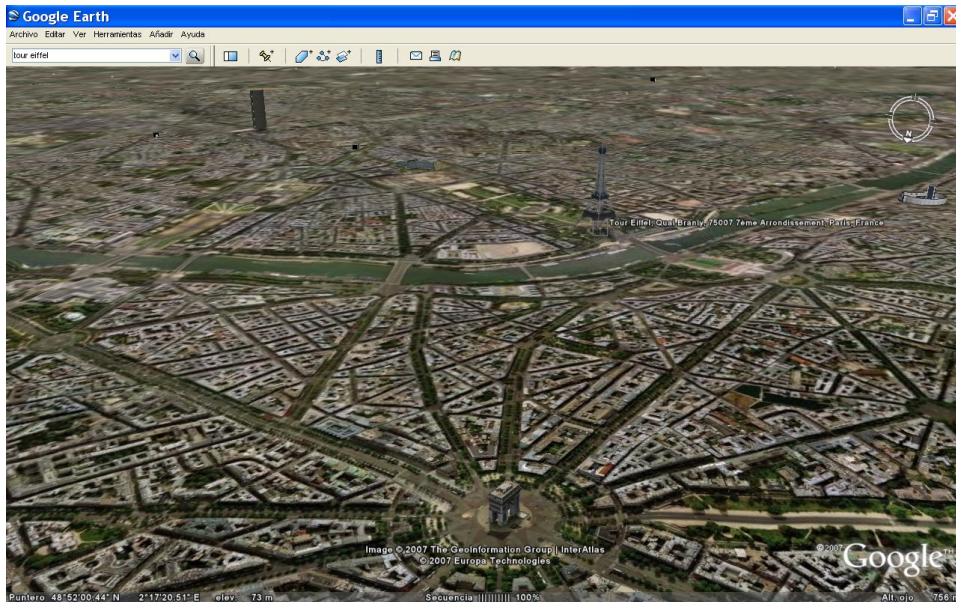


Figure 1. 3D view of Paris in Google Earth, including several 3D models of famous buildings

Although Google Earth and Google Maps are not compliant to OGC standards, their free cartography, simple but rich interface and the possibility for users to add their own geographic data, have massively popularized these tools.

Furthermore, Google Maps provides a simple JavaScript API that enables third parties to publish the location of spots on the map, and to connect to other web services through AJAX calls. An example can be seen in the Figure 2, where Google Maps is used to show the position of San Francisco buses in real time.

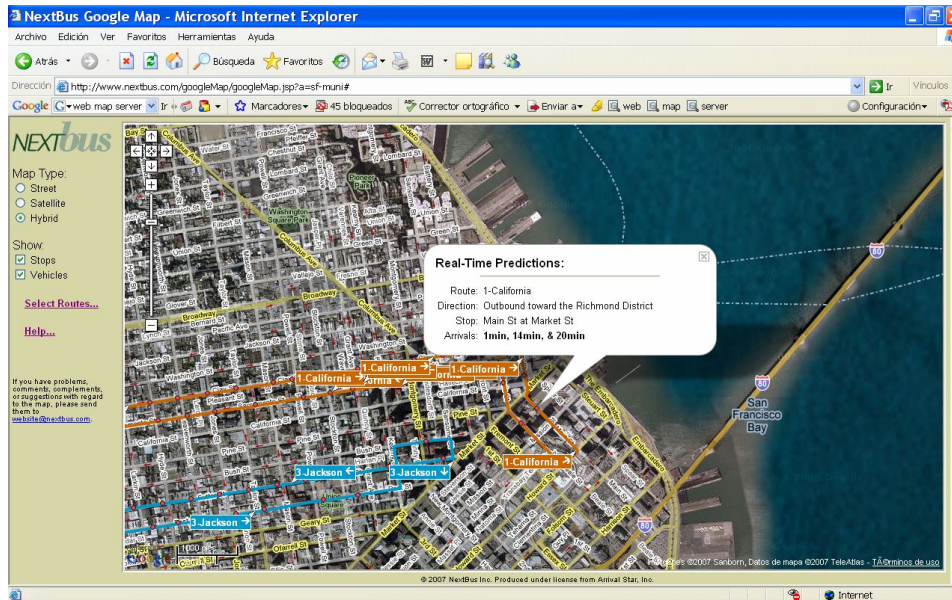


Figure 2. San Francisco buses in real time, at <http://www.nextbus.com/>

Due to its popularity and easy API, we have decided to build the first prototype of our map-enabled QTI service on Google Maps. Nevertheless, as it will be discussed later, our plans include to extend the middleware to also support OGC-compliant map servers.

#### 4. Connecting APIS and Google Maps

The connection between APIS and Google Maps is carried out through a middleware that we have developed. This middleware is the responsible for generating the Google Maps JavaScript functions to show the map. It also processes the actions of the user on the map and converts them into QTI responses that can be processed by APIS. The middleware encapsulates all the Google Maps code, and consequently the APIS engine does not need changes. Providing support for WMS would require modifications in the middleware, but not in APIS.


A map can be inserted into a question through the tag *map* in the *itemBody*, in the same way as an image is inserted. The *map* element has an attribute *src* that contains the URL of a file describing the map. This file contains the coordinates of the centre of the map (according to the WGS84 reference system) and the zoom level (1 is world-wide level and 18 is street-level). Other parameters can also be set concerning visualization aspects.

```
<Centre>
  <Latitude>40.346544</Latitude>
  <Length>-3.757324</Length>
  <Zoom>5</Zoom>
</Centre>
```

Once APIS detects a *map* tag, it calls the middleware that parses the XML file and generates the JavaScript code necessary to set the right map. This JavaScript code is inserted into the web page that presents the question. For some questions this is the whole process since only showing the map is required, with no interaction, exactly like an image. Figure 3 shows an example. Note that, unlike a still image, an interactive map enables the user to freely put the displayed area in context by using zoom controls or by moving the map to explore neighbouring areas. Nevertheless, the teacher can lock this possibilities when necessary, through some parameters in the XML file.



Look at the map and answer the question below;



Which is the peninsula that you can see on the map:

- Italian Peninsula
- Iberian Peninsula
- Balkan Peninsula
- Kola Peninsula

submit

Figure 3. Example of a simple choice question showing a map from Google Maps

However, we are mainly interested on more complex questions that have to be answered by the user by interacting with the map. As an example, we can see the question in Figure 4, where the user is required to click on the largest Catalan city.

Look at the map and answer the question below;



Which is the largest city in Catalonia?

submit

Figure 4. Example of a simple choice question requiring the user to interact with the map

We can observe that, like in the example of the Iberian Peninsula above, this is a simple choice question, with four choices and just one valid. However the four options are not explicitly included in the text below the map, but are included as markers on the map. The user is required to click on the markers to answer the question, providing a more natural interaction.

The QTI file contains the information indicating that the right answer is Barcelona, in the *responseDeclaration* element. However, the four choices are not explicitly declared in the *choiceInteraction* element, since they should not appear in the text below the map. Instead, choices are declared as *GMapInteraction*, which indicates to APIS that the interaction for this question is defined in the map description file and consequently, passing the responsibility to the middleware. The middleware has to generate the JavaScript functions that handle users clicks, and have to convert these clicks on a string with the selected choice (Barcelona, Girona, Tarragona or Lleida) that will be sent back to APIS. APIS will then check whether the selected choice is the right answer (Barcelona) or not, in the normal way. Note that the only change introduced to APIS is the detection of the *GMapInteraction* type of choice, while all the geographic logic is managed by the middleware through JavaScript functions.

This method enables the use of maps in other types of QTI interactions. For instance, the following image shows an *orderInteraction*, where the user is prompted to order the four Catalan capitals according to their population.



Figure 5. Example of an order interaction requiring the user to interact with the map

Furthermore, we can define more complex ways of interacting with the map. In the following example, the user is prompted to click on Italy, and there is an invisible polygon defining the area that is considered as the right answer.



Figure 6. Example of a new type of question: *PointIntoPoligon* map interaction

Note that although this can be modelled as a simple choice question with two possibilities: *Italy* (right choice) or *outside Italy* (wrong choice), we have to define how this has to be handled in Google Maps. This is an example of what we have called *map interactions*, which are different than question interactions. A map interaction defines how a given answer and the solution are processed (through spatial operations) to validate the correctness of the answer, and it is defined in the XML file describing the map. In this case, the map interaction is a *PointIntoPolygon*, and has two parameters: the polygon defining Italy, and a Boolean value indicating that the polygon is kept hidden. Note that the middleware, and not APIS, is the responsible of evaluating the map interaction, i.e. checking whether the user has clicked inside the polygon that defines Italy or not.

We have defined a list of other types of map interactions, that basically correspond to the different topological relations. In the following example, where the user is prompted to draw the border line between Spain and Portugal, the map interaction is a *LineIntoBufferedLine*. In this case, the solution is a line that is widened (through a buffer spatial operator) to a certain distance, for instance 50 km. This distance is set in the XML file. The middleware has to check whether the line that the user has drawn is inside the buffered line or not.





Figure 6. Example of *LineIntoBufferedLine* map interaction

## 5. Conclusions and future work

Maps are an important element in Geography education. An eLearning approach should not skip both the necessity and the potential of the use of maps in Geography learning. In particular, assessment can be improved if interactions with maps are supported, enabling users to click or sketch on maps.

We have presented a middleware-based approach to connect the APIS QTI engine and Google Maps. This approach enables the definition of mapping services for assessment that involve just some minor changes to the APIS QTI engine.

The use of interactive maps introduce the possibility of new ways of interaction with questions. The user can be prompted to click on markers instead of selecting text options, providing a more natural way of interacting with geographic information. Furthermore, the user can also be asked to draw sketches (points, lines or polygons) as his/her response to a question, introducing the concept of map interactions that define how the correctness of the answer is processed. This way, new types of queries not covered by QTI can be defined.

Our plans for future work related to this paper include three main lines. The first one consists in the development of a new middleware to support the connection to WMS servers, such as MapServer. A second line is related to a new editor for QTI 2.1 that we are currently developing. We plan to include a module enabling authors to graphically create map-enabled questions and map interactions. Finally, as a third line, we would like to explore the definition of new types of questions, with other map interactions including support for more geometries and spatial operations.

In addition, from our point of view, an eLearning environment should support the combination of different standards and specifications to provide richer contents and interactivity. We think that our work on enhancing assessment tests with maps has a more general aim, presenting an approach for integrating different specifications in an eLearning (assessment in this case) environment. In this line, we plan to integrate other services in our IMS QTI engine, following the middleware-based approach we have used here, to enable a collaborative and multimedia assessment. On the other hand, we also plan to extend our current middleware to be integrated in an IMS Learning Design (LD) [11] engine, again as a first exploratory example of combining different specifications with LD.

## Acknowledgment

This work has been partially sponsored by the TENCompetence Integrated Project that is funded by the European Commission's 6<sup>th</sup> Framework Programme, priority IST/Technology Enhanced Learning. Contract 027087 (<http://www.tencompetence.org>)

## References

- [1] IMS Global Learning Consortium: *Question & Test Interoperability Specification v2.0/v2.1*. <http://www.imsglobal.org/question/> (last access: May 2007)
- [2] APIS project: *Assessment Provision through Interoperable Segments*. <http://sourceforge.net/projects/apis/> (last access: May 2007)
- [3] OKI Project: *Open Knowledge Initiative*. <http://www.okiproject.org/> (last access: May 2007)
- [4] IMS Global Learning Consortium: *General Web Services Specification*. <http://www.imsglobal.org/gws/> (last access: May 2007)
- [5] Wills, G.; Davis, H.; Chennupati, S.; Gilbert, L.; Howard, Y.; Jam, E. R.; Jeyes, S.; Millard, D.; Sherratt, R. and Willingham, G. (2006) R2Q2: Rendering and Responses Processing for QTIV2 Question Types. In Proceedings of 10th International CAA Conference, Loughborough University, UK.
- [6] R2Q2 project: Rendering and Response processing services for QTIV2 questions. <http://www.r2q2.ecs.soton.ac.uk/> (last access: May 2007)
- [7] Blat, J.; Navarrete, T.; Moghnieh, A. and Batlle Delgado, Helena. A QTI Management System for Service Oriented Architectures. TENCompetence Open Workshop on Service Oriented Approaches and Lifelong Competence Development Infrastructures. Manchester, 11-12 January 2007
- [8] Open Geospatial Consortium: *Web Map Service Implementation Specification*. <http://www.opengeospatial.org/standards/wms/> (last access: May 2007)
- [9] University of Minnesota: MapServer. <http://mapserver.gis.umn.edu/> (last access: May 2007)
- [10] International Organization for Standardization: *Geographic information - Web Map server interface standard (ISO 19128:2005)*. 2005.
- [11] IMS Global Learning Consortium: *Learning Design Specification*. <http://www.imsglobal.org/learningdesign/> (last access: May 2007)



## ANNEX 2:

### Joc de Proves

---

*Tipus de pregunta Choice:*

#### Item :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example adapted from the PET Handbook, copyright University of Cambridge ESOL Examinations --
>
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0 imsqti_v2p0.xsd"
  identifier="ClickChoiceMap" title="Unattended Luggage" adaptive="false" timeDependent="false">
  <responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
    <correctResponse>
      <value>Barcelona</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
    <defaultValue>
      <value>0</value>
    </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p>Look at the map and answer the question below; </p>
    <p>
      <map src="GoogleMaps.xml"/>
    </p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
      <prompt>Which is the largest city in Catalonia?</prompt>
      <simpleChoice identifier="GMapInteraction"></simpleChoice>
    </choiceInteraction>
  </itemBody>
  <responseProcessing
    template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>
```

#### Descripció del mapa (GoogleMaps.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<GoogleMaps>
  <InteractionType>Choice</InteractionType>
  <Center>
    <latitude>41.38</latitude>
    <length>2.18</length>
    <zoom>7</zoom>
  </Center>
  <Respostes>
    <OptionMarker>
      <image>"C:/PFC/Apis/apis_viewer/src/MiddleWare/OpcioA.gif"</image>
      <info>Girona</info>
      <latitude>41.981219</latitude>
      <length>2.823443</length>
    </OptionMarker>
    <OptionMarker>
      <image>"C:/PFC/Apis/apis_viewer/src/MiddleWare/OpcioB.gif"</image>
      <info>Barcelona</info>
      <latitude>41.387918</latitude>
      <length>2.169929</length>
    </OptionMarker>
    <OptionMarker>
      <image>"C:/PFC/Apis/apis_viewer/src/MiddleWare/OpcioC.gif"</image>
    </OptionMarker>
  </Respostes>
</GoogleMaps>
```

```

        <info>Tarragona</info>
        <latitude>41.133676</latitude>
        <length>1.244888</length>
    </OptionMarker>
    <OptionMarker>
        <image>"C:/PFC/Apis/apis_viewer/src/MiddleWare/OpcioC.gif"</image>
        <info>Lleida</info>
        <latitude>41.60787</latitude>
        <length>0.632744</length>
    </OptionMarker>

</Respostes>

</GoogleMaps>

```

## Tipus de pregunta PointIntoPolygon:

### Item:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- This example adapted from the PET Handbook, copyright University of Cambridge ESOL Examinations --
>
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0_imsqti_v2p0.xsd"
  identifier="LineArea" title="Unattended Luggage" adaptive="false" timeDependent="false">
  <responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
    <correctResponse>
      <value>Correcte</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
    <defaultValue>
      <value>0</value>
    </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p>Look at the map and answer the question below </p>
    <p>
      <map src="GoogleMaps9.xml"/>
    </p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
      <prompt>Draw the frontier between Spain and Portugal</prompt>
      <simpleChoice identifier="GMapiInteraction"></simpleChoice>
    </choiceInteraction>
  </itemBody>
  <responseProcessing
    template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>

```

### Descripció del mapa:

```

<?xml version="1.0" encoding="UTF-8"?>
<GoogleMaps>
  <InteractionType>LinesInPolygon</InteractionType>
  <Center>
    <latitude>40.346544</latitude>
    <length>-3.757324</length>
    <zoom>5</zoom>
  </Center>
  <Polygon>
    <Vertex>
      <ID>1</ID>
      <latitude>42.413318</latitude>
      <length>-9.144745</length>
    </Vertex>

```



```
<Vertex>
  <ID>2</ID>
  <latitude>42.238177</latitude>
  <length>-6.091919</length>
</Vertex>
<Vertex>
  <ID>3</ID>
  <latitude>36.803787</latitude>
  <length>-6.706467</length>
</Vertex>
<Vertex>
  <ID>4</ID>
  <latitude>36.712467</latitude>
  <length>-9.206543</length>
</Vertex>
<Vertex>
  <ID>5</ID>
  <latitude>42.413318</latitude>
  <length>-9.144745</length>
</Vertex>
</Polygon>
</GoogleMaps>
```