# Accès et personnalisation du contenu multimédia dans un véhicule

Stéphane Turlier

## To cite this version:

Stéphane Turlier. Accès et personnalisation du contenu multimédia dans un véhicule. Informatique mobile. Télécom ParisTech, 2011. Français. NNT : . pastel-00683823

## HAL Id: pastel-00683823
## https://pastel.hal.science/pastel-00683823

Submitted on 29 Mar 2012

STÉPHANE TURLIER

# ACCÈS ET PERSONNALISATION DU CONTENU MULTIMÉDIA DANS UN VÉHICULE

# ACCÈS ET PERSONNALISATION DU CONTENU MULTIMÉDIA DANS UN VÉHICULE

STÉPHANE TURLIER

Thèse de doctorat ès sciences mention « informatique et réseaux »

Département Communications Multimédia d'Eurécom
École Doctorale d'Informatique Télécommunications et Électronique de Paris

mai 2011

Stéphane Turlier : *Accès et personnalisation du contenu multimédia dans un véhicule* .
Thèse de doctorat ès sciences mention « informatique et réseaux » ,
soutenue le 30 mai 2011 à Sophia-Antipolis.

PRÉSIDENTE DU JURY :
Professeur Isabelle Demeure, TELECOM ParisTech
RAPPORTEURS :
Professeur Françoise Prêteux, MINES ParisTech
Professeur Gerhard Rigoll, Technische Universität München
DIRECTEUR DE THÈSE :
Docteur Benoit Huet, EURÉCOM
EXAMINATEUR :
Docteur Hans-Jörg Vögel, BMW

À mes deux grand-pères, Charles Turlier et Hubert Dubray,
ingénieurs dans l'industrie française.

# RÉSUMÉ

L'arrivée récente de plateformes véhiculaires connectées à internet permet la diffusion de contenus d'infodivertissement en flux poussés et tirés pour les conducteurs et les passagers en situation de mobilité d'une manière comparable aux appareils de communication nomades actuels. Toutefois, la voiture constitue un terminal d'accès très différent d'un téléphone portable que ce soit en termes de caractéristiques techniques, mais aussi en matière d'usages.

Cette thèse aborde le sujet de la fourniture personnalisée du contenu multimédia pour les automobilistes. Une étude des caractéristiques techniques des plateformes d'infodivertissement, des types de contenus et de métadonnées nous permet dans un premier temps de cerner les contraintes d'une architecture de fourniture individualisée pour un véhicule. La mise en perspective de ces contraintes nous permet d'établir une architecture de fourniture de contenu à la demande, implémentée dans un prototype.

Nous abordons ensuite le problème de la personnalisation suivant deux axes complémentaires : D'une part, la personnalisation active qui fait intervenir une interface homme machine multimodale que nous étudions dans le cadre spécifique d'un navigateur de bibliothèque musicale en ligne permettant de créer facilement des listes de lectures multicritères dans un véhicule ; et d'autre part, la personnalisation passive du contenu qui fait intervenir une modélisation du contexte de l'utilisateur. Nous discutons de la répartition de composants fonctionnels permettant cette personnalisation passive et construisons une architecture répartie prenant en compte la définition individualisée de préférences contextuelles et son intégration dans l'architecture multimédia présentée plus haut. Les différentes solutions proposées sont enfin évaluées selon des méthodes expérimentales faisant intervenir des utilisateurs et des méthodes dites expertes.

# ABSTRACT

The recent advent of connected vehicle platforms permits the distribution of infotainment assets to drivers and passengers with pulled and pushed workflows in a compoarable manner to current mobile handsets. However, vehicles differ technically from mobile phones in terms of capability and in terms of usage.

This thesis tackles the subject of personalised media delivery to motorists. We first study the technical characteristics of vehicle infotainment platforms, media assets and metadata in order to identify the requirements of a media delivery architecture for a vehicle. Based on those constraints, we have specified a media on-demand framework which has been developped in a prototype.

Afterwards, we tackle the topic of personalisation in light of two complementary point of views: On the one hand, the driver can process active personalisation when using a proper human machine interface. We present a music browser for online libraries which allows the creation of multicriteria playlists while driving. On the other hand, we analyse passive personalisation which makes use of the driving's context. We discuss the repartition of the functional components and build up a distributed architecture which takes into account individual context preferences and their integration in the multimedia architecture that we have formerly presented. Eventually, the different solutions are evaluated according to experimental and expert methods.

## ZUSAMMENFASSUNG

Die jüngste Einführung neuer vernetzter Fahrzeugplattformen erlaubt den Vertrieb von Infotainment-Inhalten mit so genannten ‚pushed' und ‚pulled' Prozessen für Fahrer und Beifahrer so wie es für mobile Geräte bereits erfolgt. Die eingebetteten Systeme des Fahrzeugs unterscheiden sich jedoch von mobilen Endgeräten in ihren technischen Charakteristiken sowie in ihrer Nutzung.

Diese Doktorarbeit befasst sich mit dem personalisierten Vertrieb von Infotainment-Inhalten für Fahrzeug-Nutzer. Eine initiale Studie der verschiedenen technischen Charakteristiken einer Infotainment-Plattform, Medientypen und Metadaten erlaubt es uns, die Anforderungen an eine angepasste Architektur zum Vertrieb vernetzter Medien für Fahrzeuge zu identifizieren. Die so spezifizierte Architektur wird anhand eines Prototyps erläutert.

Anschließend wenden wir uns dem Thema der Personalisierung zu, indem wir zwei komplementäre Ansätze verfolgen: Einerseits untersuchen wir die aktive Personalisierung unter Verwendung von Mensch-Maschine-Interaktion. Im speziellen Fall die eines Musikbrowsers zur Erzeugung multikriterieller Playlisten im Fahrzeug; anderseits die passive Personalisierung, die ein Kontext-Modell benötigt. Wir diskutieren die Verteilung von funktionalen Komponenten, die diese passive Personalisierung ermöglichen und entwerfen ein Architekturmodell, welches individuell spezifizierte Präferenzen berücksichtigt und, erörtern die Integration in die zuvor diskutierte Medien-Vertriebs-Architektur.

Schließlich werden die verschiedenen, implementierten Lösungen mithilfe der experimentellen Methode der Stichproben-Untersuchung so wie der Expertenmethoden evaluiert.

# REMERCIEMENTS

> *« C'est plutôt par l'estime de nos propres sentiments que nous exagérons les bonnes qualités des autres, que par l'estime de leur mérite ; et nous voulons nous attirer des louanges, lorsqu'il semble que nous leur en donnons. »*
> — François, duc de La Rochefoucauld, *Réflexions ou Sentences et Maximes morales, 1664*

Malgré les réserves de de La Rochefoucauld, je me dois de saluer les nombreuses personnes qui ont permis la réalisation de cette thèse.

## ÉCRIRE UNE THÈSE, C'EST AVANT TOUT UNE HISTOIRE DE RENCONTRES

Je dois en premier lieu adresser ma plus grande gratitude à Hans-Jörg Vögel qui m'a conseillé de faire cette thèse.

Je souhaite également remercier mon directeur de thèse Benoit Huet pour sa flexibilité, son indulgence et surtout pour la motivation qu'il a su me redonner à des moments cruciaux de mon travail. Je mesure encore mal, les trésors de patience dont il a du faire preuve, alors qu'il devait diriger une thèse qui se déroulait à quelques milliers de kilomètres de son laboratoire avec un étudiant au tempérament difficile. Je remercie également le département MMk de la TUM pour m'avoir invité à ses séminaires.

Je voudrais saluer le travail des rapporteurs Françoise Prêteux et Gehrard Rigoll pour leurs analyses pointues et Isabelle Demeure pour avoir accepté de diriger ce jury franco-allemand.

Je me dois de remercier les structures qui m'ont accueilli :

- D'abord chez BMW Forschung und Technik pendant plus de trois années, je dois saluer David Bloom du Technology Office de Palo Alto avec qui j'ai travaillé sur mon premier projet, et qui a depuis la Californie suscité de nombreuses idées de cette thèse. Les doctorants du département IT Drive de Munich, en particulier Witold Drytkiewicz avec qui j'ai partagé pratiquement l'intégralité de mon temps de thèse. Je voudrais aussi exprimer envers deux personnes qui ont accompagné la fin de cette thèse chez BMW, Dominik Schnieders qui a repris la direction de l'équipe architecture et Thomas Helbig qui dirigeait le projet AENZU, ma reconnaissance, toute proportionnelle à leur diligence et leur disponibilité.

- Ensuite le département communications multimédia dirigé par Bernard Merialdo, pour avoir accepté de me prendre en thèse et Raphael Troncy pour les nombreux liens qu'il m'a fourni pour illustrer des détails intéressants du web sémantique.

Enfin, je souhaite témoigner ma reconnaissance aux formidables étudiants avec lesquels j'ai eu la chance de travailler chez BMW (et qui ont je l'espère pris du plaisir à travailler avec moi) : Olena, Marc, Clemens, Sascha et Robert.

### écrire une thèse, c'est aussi une affaire de tempérament

Je souhaiterais remercier deux institutions qui ont eu un rôle décisif au cours de toutes mes études qui s'achèvent. Le collège Barral (ancien petit séminaire de Castres) et l'Institut National des Télécommunications.

### écrire une thèse, c'est parfois une question de folie

Je voudrais remercier mes parents pour leur soutien indéfectible et pour avoir été d'importants contributeurs financiers à cette thèse ; Le Doktorandenkreis (association des doctorants de BMW) pour les réunions d'information qu'ils planifient consciencieusement tous les mois et les nombreuses sorties qu'ils ont organisées ; Mes amis de Munich pour m'avoir changé les idées si souvent : Maximilian, Stephan, Alexey, Charlotte, Martina ; de Paris Zeh-Ty et Julien pour m'avoir hébergé au pied levé lors de mes quelques virées parisiennes improvisées ; de l'ENST pour m'avoir rendu visite pendant mon exil Émilie, Maya et Vincent ; ceux de Sophia-Antipolis : Sebastiano, Giuseppe. Je voudrais aussi saluer tout particulièrement Daniel avec qui j'ai passé quelques nuits à chatter sur skype alors que je terminais les illustrations de ce manuscrit pendant qu'il écrivait lui même un mémoire de musicologie.

### écrire une thèse, c'est enfin un problème de logistique

Je voudrais ici saluer la marque Railey qui a conçu un vélo avec lequel j'ai pu parcourir pas loin de 2800 km en un peu plus de trois ans, dont 400 dans les deux derniers mois alors que j'achevais la rédaction de ce manuscrit à la Bayrische Stadt Bibliothek. Du bureau de l'Hanauerstraße au FIZ, en passant par la BMW Hochhaus et la TUM dans l'Arcistraße, j'ai parcouru une bonne partie de Munich en vélo. Je n'ai pas compté le nombre de chambres à air que j'ai du changer après avoir roulé sur les obstacles les plus improbables que l'on puisse rencontrer dans une ville du vingt-et-unième siècle, la robustesse de ce VTT qui roule encore parfaitement, n'est sans doute pas complètement étrangère au succès de cette entreprise.

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## LISTINGS

# ACRONYMS

---

| | |
|---|---|
| AM | amplitude modulation |
| BiM | binary format for xml |
| CAN | controller area network |
| CDN | content delivery network |
| CIC | car infotainment computer |
| CID | car infotainment display |
| CF | collaborative filtering |
| CRM | customer relationship management |
| DAB | digital audio broadcast |
| DHT | distributed hash tables |
| DRM | digital rights management |
| EBU | european broadcasting union |
| EDGE | enhanced data rates for GSM Evolution |
| EPG | electronic program guide |
| FM | frequency modulation |
| FOAF | friend of a friend |
| GALA | geschwindigkeitsabhängige Lautstärkeanpassung |
| GPRS | general packet radio service |
| HiFi | high fidelity |
| HMI | human machine interface |
| IPTC | international press communication council |
| ISP | internet service provider |
| IVI | in-vehicle infotainment |
| LVDS | low-voltage differential signaling |
| LTE | long-term evolution |
| MIME | multipurpose internet mail extensions |
| MIR | media indexing and retrieval |

| | |
|---|---|
| MOST | media oriented systems transport |
| MPEG | motion picture expert group |
| OSI | open systems interconnection |
| OWL | ontology web language |
| QbH | query by humming |
| RDS | radio data system |
| RDF | resource description framework |
| SMPTE | society of motion picture and television engineers |
| T-DMB | digital media broadcast |
| UMTS | universal mobile telecommunications system |
| URL | uniform resource location |
| URI | uniform resource identifier |
| XML | extended markup language |
| XSD | XML schema definition |
| W3C | world wide web consortium |

# Part I

# RÉSUMÉ DE LA THÈSE EN FRANÇAIS

# 1

# ACCÈS ET PERSONNALISATION DU CONTENU MULTIMÉDIA DANS UN VÉHICULE[0]

## 1.1 introduction

Quelques jours après leur victoire aux élections présidentielles américaines de 1992, Bill Clinton et son administration se penchèrent sur les moyens de développer les médias connectés en tant qu'industrie. A l'époque, internet n'existait pas tel que nous le connaissons aujourd'hui. Il s'agissait d'un réseau d'ordinateurs reliant des universités et des instituts de recherche. Dans le grand public, la télévision était encore le média dominant et après le boom des réseaux câblés des années 70 et 80, le satellite se répandait en zone rurale pour gagner de nouveaux téléspectateurs. Les gens achetaient des CDs de leurs chanteurs favoris, et des cassettes VHS du dernier film dont ils avaient entendu parler. Rechercher, acheter, regarder, et diffuser le contenu était laborieux et parfois décourageant.

L'initiative prise par le jeune vice-président Al Gore, devait changer la face de l'industrie et a encore des conséquences vingt années plus tard sur la manière dont nous consommons les médias. De nouvelles manières de consommer les médias ont été introduites par le commerce électronique: le média à la demande. Pendant ce temps, les 'autoroutes de l'information' sont devenues mobiles avec le développement de la téléphonie mobile et s'intègrent aujourd'hui au moyen de locomotion individuelle : la voiture.

## 1.2 position du problème: les applications d'infodivertissement doivent être adaptées aux contraintes des plateformes véhiculaires et de mobilité

L'objectif de cette thèse est d'apporter une contribution dans le domaine du multimédia appliqué à des scénarios automobiles. On substitue le terme infodivertissement au terme multimédia dans le domaine des véhicules et la majorité des cas d'utilisation de contenu multimédia présentés dans cette thèse concernent du contenu de divertissement audio et en particulier de la musique, puisqu'il s'agit du média le plus consommé dans les véhicules. Cependant, rendre le contenu d'internet disponible dans les véhicules implique des adaptations à différents niveaux.

---

[0] Nous donnons ici une traduction partielle de la thèse initialement écrite en anglais à la demande de BMW afin de nous conformer à la loi française recommandant l'usage du français pour les examens, les mémoires et les thèses (article L121-3 du code de l'éducation). Les explications détaillées des raisonnements des différents chapitres ainsi que les illustrations et les résultats ne sont donnés que dans la partie rédigée en anglais de ce mémoire. Les sources données dans la bibliographie (*cf. supra p.* 211) sont valables pour l'ensemble de la thèse, toutefois, elles ne sont mentionnées dans le détail que dans la partie rédigée en anglais.

### *Adaptation de l'architecture*

Les véhicules sont des systèmes complexes qui impliquent des sous-systèmes avec de hauts niveaux de sécurité et ont de ce fait des cycles de développement relativement long; en revanche, l'écosystème des médias connectés est caractérisé par un changement rapide et continu des services. L'architecture d'un système de distribution de média pour les véhicules doit prendre en compte cette nécessité afin de proposer une solution robuste tout en maintenant flexibilité et évolutivité.

### *Adaptation du design*

Consommer des médias dans un véhicule est très nettement différent de l'utilisation classique d'un téléviseur, d'un ordinateur ou d'un baladeur. Les aspects d'ergonomie impliquent non seulement des modalités utilisées pour les interactions, mais également l'architecture elle-même du logiciel en terme d'efficacité et de transparence pour l'utilisateur.

### *Adaptation de la sélection de médias*

Avec la charge croissante de médias qui sont mis à disposition pour l'utilisateur, leur sélection peut introduire une importante charge cognitive. Celle-ci peut être réduite en assistant et en prévoyant les choix opérés par l'utilisateur, en fonction du contexte. En effet, l'adaptation du design des applications n'est pas toujours suffisante et la sélection des médias doit être adaptée à la situation de conduite.

### *Nouveaux scénarios, nouveaux concepts, nouvelles solutions*

En réalité, l'intégration automobile de services multimédias préexistants implique bien plus qu'une simple adaptation. Ce sont de nouvelles architectures de services pour de nouveaux cas d'utilisation qui doivent être créés:

- Une agrégation efficace et un préchargement des métadonnées: La diversité des informations concernant les médias en ligne est une clé pour le développement de méthodes de sélection. Les métainformations venant de sources diverses doivent être agrégées de manière cohérente et mise à disposition du client multimédia mobile (le véhicule) qui a des ressources limitées.
- Une distribution du raisonnement contextuel : Aider l'utilisateur dans son processus de décision, ou prévoir ses intentions nécessitent un modèle de calcul, mais également une architecture permettant le déploiement d'entité de raisonnement dans les différentes entités (fournisseur de contenu, récepteur) et un modèle de persistance des préférences permettant leur dissémination.

### *Le prototypage*

Non seulement cette thèse propose de nouvelles solutions au problème suscité d'un point de vue technique, elle a également pour objet de fournir des contributions scientifiques.

Tous les concepts présentés dans cette thèse ont été développés avec une étude détaillée de l'état de l'art (recherche incrémentale) et sont basés sur des discussions avec des spécialistes des différents domaines de recherche qu'impliquent le sujet (recherche exploratoire). La plus grande partie des recherches furent conduites dans l'équipe de recherche en architecture chez BMW Research and Technology et les méthodes de recherche furent adaptées aux méthodes de travail de l'équipe industrielle habituée à concevoir des 'démonstrateurs'. Cela consiste à réaliser des 'proof of concepts' (implémentation d'un prototype illustrant les concepts théoriques) pour démontrer la faisabilité technique de l'architecture étudiée.

## 1.3 PLAN DE LA THÈSE[1]

Les deux premiers chapitres introduisent les différents domaines impliqués dans la thèse: Scénarios d'utilisation du multimédia dans les véhicules embarqués, recommandation de contenu à la demande.

- *Chapitre 2 – La consommation mobile de média dans les voitures : Des paradigmes physiques et virtuels.* Ce chapitre définit le problème de la consommation mobile de médias comme un problème centré sur l'utilisateur. Nous y analysons les différentes étapes de la sélection de médias à la demande et identifions les concepts clés de la personnalisation pour l'utilisateur. Nous présentons également les services multimédias qui sont apparus avec le contenu en ligne. Nous introduisons enfin les caractéristiques techniques des véhicules en tant que clients multimédias dans un monde connecté.

- *Chapitre 3 – La révolution numérique: les métadonnées et la recommandation.*

  Ce chapitre explique comment les métadonnées qui sont apparues avec la numérisation du contenu peuvent être utilisées afin d'aider l'utilisateur dans son usage des médias. Des généralités sur les métadonnées sont suivies par une analyse des flux d'extraction et de production des métadonnées. Enfin, nous présentons la recommandation multimédia et discutons la faisabilité de son intégration dans un scénario multimédia.

Les chapitres 4, 5 et 6 répondent aux questions centrales de notre problème: Comment acheminer du contenu multimédia dans un véhicule ? Comment rendre possible la personnalisation de ce contenu ?. Nous répartissons l'analyse en deux sous-problèmes: la personnalisation active et la personnalisation passive.

- *Chapitre 4 – La fourniture de contenu à la demande, architecture du système Personal Radio.* Ce chapitre explique le choix des architectures web multicouches pour la fourniture de services à la demande. Nous commençons par comparer les services en diffusion, centralisée avec routage d'informations et point à point. Ensuite nous présentons l'implémentation de l'architecture *Personal Radio* qui combine un client riche avec un service d'abstraction. Nous illustrons la flexibilité de cette architecture avec une étude d'un

---

[1] il s'agit ici du plan des chapitres de la thèse rédigée en anglais, le résumé en français reprend toutefois exactement la même organisation

mécanisme d'envoi de publicité personnalisée lorsque notre architecture peut-être couplée à un système de customer relationship management (CRM).

- *Chapitre 5 – La personnalisation interactive du contenu, naviguer au travers des recommandations musicales en conduisant.* Ce chapitre est une analyse des aspects ergonomiques d'une application d'infodivertissement au travers d'une étude prototype : un navigateur musical. Nous analysons les différentes modalités qui peuvent être combinées pour les entrées/sorties de l'utilisateur ainsi que les différentes alternatives architecturales pour fournir à l'utilisateur des recommandations et l'aider dans son choix. Nous présentons ensuite l'implémentation d'un navigateur au travers de ses composants fonctionnels et techniques.

- *Chapitre 6 – La personnalisation contextuelle passive du contenu.* Ce chapitre présente une alternative à la solution précédente. Au lieu de demander à l'utilisateur d'interagir avec l'application, nous pouvons utiliser le contexte pour en déduire automatiquement ses souhaits. Nous commençons par analyser le problème des systèmes sensibles au contexte et les différents types d'architectures en ce qui concerne l'accès aux données contextuelles, le raisonnement et la transposition à une application multimédia. Le chapitre décrit un système basé sur la logique floue que nous avons implantée au sein de l'architecture de *personal radio* et qui offre des solutions aux problèmes suivants : persistance des préférences contextuelles des utilisateurs, la sélection de médias distants basées sur le contexte, le rendu géolocalisé de média.

Les deux derniers chapitres évaluent les différentes solutions proposées dans les précédents chapitres et tirent les conclusions de notre travail afin de dégager de futures directions de recherche.

- *Chapitre 7 – Évaluation des concepts.* Après avoir revu la littérature très riche concernant les méthodes d'évaluation concernant les méthodes d'évaluations d'applications véhiculaires, nous présentons une étude qualitative de notre navigateur musical présenté dans le chapitre 5 basé sur la méthode Attrakdiff. Nous présentons également des mesures faites sur l'architecture. Enfin, nous conduisons une étude heuristique de la solution de personnalisation contextuelle du chapitre 6.

- *Chapitre 8 – Conclusion et recherches futures.* Ce chapitre tire les conclusions et récapitule les contributions de notre travail afin d'identifier des axes de recherches dont nous identifions un potentiel pour le futur

# 2

# DÉVELOPPEMENT ET RÉSULTATS : PERSONNALISATION DE L'INFODIVERTISSEMENT

## 2.1 LA CONSOMMATION MOBILE DE MÉDIA DANS LES VOITURES : DES PARADIGMES PHYSIQUES ET VIRTUELS

La consommation des médias guide aujourd'hui la conception de la plupart des produits vendus dans l'électronique grand public et les autoradios n'y font pas exception. L'offre elle-même a beaucoup changé avec la numérisation des contenus. Ce chapitre éclaire les tendances actuelles de la consommation des médias et décrit la relation avec notre problème : la personnalisation de l'infodivertissement dans l'expérience véhiculaire de l'utilisateur. Nous déduisons d'abord d'un scénario les prérequis fonctionnels d'une solution de personnalisation. Ensuite, nous analysons le processus de consommation et étudions systématiquement les différentes étapes de l'expérience utilisateur. Ensuite, nous analysons les différents types de services qui sont apparus dans la dernière décennie et la place du véhicule dans la chaîne de personnalisation.

### *Analyse du scénario de consommation de contenu multimédia dans un véhicule*

Alice est mariée et a deux enfants, elle habite dans une grande ville et son trajet quotidien jusqu'à son travaille pour un journal dure de 20 à 30 minutes pour l'aller et environ la même chose pour le retour. Le temps de trajet tout comme l'heure exacte du départ peuvent varier selon son emploi du temps. Alice aime écouter de la musique lorsqu'elle conduit, mais doit aussi rester à la page en matière de nouvelles économiques et politiques pour son travail. C'est pourquoi elle écoute les informations lorsqu'elle conduit le matin et préfère de la musique lorsqu'elle rentre le soir. Alice aime partager ses découvertes musicales avec ses amis et ses collègues de travail. Alice appartient à la population active comme une grande majorité de personnes possédant un véhicule et est donc une cible de choix pour les publicitaires. De ce scénario nous pouvons identifier les besoins fonctionnels suivants :

- Un accès à la demande : L'accès au contenu à la demande est une des clés du scénario que nous venons de présenter. Alice a un emploi du temps qui change, mais elle doit pouvoir accéder au contenu qu'elle souhaite à tout moment. Par ailleurs, les goûts musicaux spécifiques d'Alice nécessitent la création de listes de lectures personnalisées. Les préférences d'Alice peuvent être gardées en ligne afin qu'elle puisse y accéder depuis plusieurs types d'appareils. Par exemple, elle peut continuer à interagir avec les réseaux sociaux.
- Une interface réactive : Pendant la conduite, Alice a plusieurs opportunités pour spécifier ses préférences. Elle rencontre des feux de signalisation, ou peut être retenue par des embouteillages. Une interface utilisateur réactive doit lui permettre de changer la configuration du client multimédia sans la gêner dans sa conduite.

- Utilisation du contexte : Même si l'emploi du temps d'Alice change, il est possible de trouver un motif général dans sa consommation de média et de le corréler avec ses habitudes de conduite. Elle ne veut pas le même programme suivant qu'elle conduit le matin ou l'après-midi. Le client multimédia doit donc également prendre en compte le contexte d'utilisation pour adapter les choix qu'il fait dans le personnalisation des listes de lecture.

### *La mobilité et l'infodivertissement*

L'évolution des moyens de transport façonne les sociétés. Aujourd'hui les moyens de transport individuels ont explosé, mais l'usage de l'infodivertissement revêt des aspects positifs et négatifs. Dibben and Williamson [2007] ont souligné dans une méta-étude que les accidents de la circulation impliquaient souvent l'usage d'un système d'infodivertissement (comme une radio). Par ailleurs, dans la même étude, ils notent que l'infodivertissement est souvent mentionné comme un moyen d'atténuer la sensation de fatigue du conducteur. Nous avons donc là deux éléments importants de notre étude. Nous devons adapter la sélection du contenu aux préférences de l'utilisateur, mais également veiller à ce que l'utilisation de notre système de contenu à la demande ne soit pas accidentogène.

### *Le processus de consommation du multimédia*

La consommation des médias a été étudiée par les comportementalistes dans les domaines de la sociologie et de l'économie. Le modèle de McQuail introduit les notions de facteurs de court et long termes dans le choix des médias. Pour notre étude, nous proposons de décomposer le processus de consommation de média à la demande dans les 7 étapes suivantes:

- Découverte : C'est la première étape durant laquelle l'utilisateur prend connaissance des possibilités offertes par les modalités de l'application (en particulier : visuelle, audio, haptique)
- Formulation de la requête : Elle varie d'un système à l'autre. Par exemple, il peut s'agir d'une simple pression sur une touche pour commencer à jouer, ou une requête de recherche plus complexe
- Filtrage : La plupart des systèmes de multimédia à la demande n'affichent pas cette étape à l'utilisateur. La sélection de contenu à la demande peut être basée sur la requête ou sur la connaissance préalable des préférences de l'utilisateur.
- Affichage des résultats : Lorsque les contenus sont filtrés ils peuvent être tous ou seulement en partie affichés à l'utilisateur de façon à ce que ce dernier prenne une décision.
- Sélection : Cette étape est optionnelle, mais beaucoup de systèmes de contenu à la demande permettent à l'utilisateur de sélectionner les résultats parmi les résultats du filtrage
- Rendu : Le rendu utilise uniquement les modalités audio et visuelle

- Modification : Cette étape qui est également optionnelle doit permette à l'utilisation de noter les résultats (retour de satisfaction[1]) voire de modifier la sélection sans avoir à recommencer toute la formulation de la requête.

### *Le contenu multimédia à la demande*

Les infrastructures permettant la distribution de contenu multimédia à la demande (en lieu et place de la diffusion de programmes) se sont largement répandues. On peut constater même une accélération depuis le lancement de la plateforme vidéo Youtube en 2004. À côté de leurs réseaux d'internet mobile, les fournisseurs d'accès on développé les réseaux de distribution de contenu ou CDN[2] pour accélérer l'accès aux contenus. Enfin, on assiste également au développement de l'informatique dans les nuages qui permet le déploiement de services avec le redimensionnement automatique des ressources en calcul et en débit. Katsma and Spil [2010] a réalisé une taxonomie des services de distribution de musique que nous complétons:

- Diffuseur de média étendu : Ils constituent la branche traditionnelle de la distribution de médias comme les radios. Ils proposent des services dits de rattrapage et de baladodiffusion.
- Radio internet personnalisée : Ces radios créent automatiquement un programme sur mesure suivant un profil utilisateur.
- Communauté d'échange : Ce sont des sites d'échanges où les utilisateurs peuvent échanger des commentaires, des 'tags' et se recommander mutuellement des morceaux de musique ou des vidéos.
- Vendeurs en ligne : Ils sont le pendant dématérialisé des grandes enseignes de disquaires. Ils ont mis en place différents modèles d'affaires : abonnement, paiement à l'achat, paiement à la consommation.
- Moteurs d'analyse : Ils analysent le web et les différentes sources d'informations sur les média (blogues, sites spécialisés, classement des radios, *etc.*) pour agréger des informations sur les médias

### *La place de l'infodivertissement dans l'électronique des véhicules*

L'importance du logiciel dans le développement applicatif des véhicules a littéralement explosé dans les dernières années. Aujourd'hui une voiture haut de gamme présente plus de 700 fonctions programmées par logiciel. L'infodivertissement est un des nombreux aspects de l'informatique embarquée parmi lesquels le contrôle d'injection du groupe moto propulseur, la sécurité active, l'assistance à la conduite ou encore les fonctionnalités d'habitacle. Il est caractérisé par une grande complexité des applications (entre 100 et 300 Mo de code) en comparaison avec les autres domaines (à titre de comparaison le groupe moto propulseur nécessite environ 2 Mo de code). Ceci-étant, les contraintes en matière de temps réel et de déterminisme de

---

[1] traduction de la notion de *relevance feedback*

[2] en anglais *content delivery networks*

l'infodivertissement sont beaucoup plus faibles que pour les applications critiques de la voiture.

*Les bus véhicules multiplexés*

Les différents unités de contrôle électronique d'un véhicule sont reliées par des réseaux multiplexés.

- CAN : Ce bus a été conçu par Bosch et est devenu aujourd'hui un standard de toute l'industrie automobile. Ses spécifications couvrent toutes les couches du modèle OSI, de la couche physique à la couche transport. CAN a une topologie en bus dans laquelle les UCEs communiquent par l'émission et la consommation de trames. Dans le domaine multimédia on peut utiliser CAN pour transmettre les informations de contrôle (boutons, molette) qui doivent communiquer avec l'autoradio. Toutefois il ne permet pas des débits suffisants pour transmettre les flux de média.
- MOST : Afin d'utiliser des applications multimédias qui demandent de hauts débits les constructeurs allemands ont spécifié le standard MOST. Il utilise un réseau en anneau sur fibre optique et nécessite des coûts de développement bien supérieurs au CAN.

*Le connectivité du véhicule*

Les véhicules sont capables depuis longtemps de recevoir les médias diffusés analogiques (en particulier la radio FM) et plus récemment numériques. Toutefois, l'internet mobile est une introduction récente (2007) et ouvre les perspectives du média à la demande dans les véhicules).

*Le matériel et le logiciel embarqué pour l'infodivertissement*

Comme nous l'avons vu précédemment le monde de médias connectés représente une diversité et une volatilité importante qui nécessite une réactivité sur le plan du déploiement fonctionnel de nouvelles applications. Aujourd'hui les autoradios sont de véritables PC embarqués. Les prototypes que présentons dans cette thèse, sont conçus pour un CIC-High de BMW et être piloté par un iDrive.

*L'interface utilisateur*

L'interface utilisateur joue un rôle primordial et est composée de dispositifs d'interactions couvrant les trois modalités : visuelle (écran), haptique (bouton, molette) et auditive (haut-parleurs). Les différents prototypes que nous présentons dans cette thèse s'appuient sur des plateformes de logiciel spécifiquement conçues pour le véhicule. Leur rôle est d'abstraire au mieux le matériel électronique embarqué afin de permettre une séparation des problèmes et une programmation modulaire. Par ailleurs ils doivent supporter la programmation des différentes modalités suscitées et enfin permettre de tester rapidement des applications déployées à chaud dans un véhicule sans avoir à flasher les dispositifs électroniques.

- Le Remote HMI : Le remote HMI est développé par BMW CarIT est donne une solution intéressante à la séparation des problèmes.

Il consiste en une application client-serveur basée sur un modèle similaire au système X Window. Le serveur est capable d'afficher les menus et gère les entrées de l'iDrive, il communique avec le serveur sur une liaison TCP. Le server peut charger des applications compilées sous formes de bibliothèques applicatives et déclarées par un manifestent ; elles contiennent une la logique de l'application. L'avantage de ce système est l'existence d'un environnement de simulation sur Windows qui permet de développer les applications sur un PC puis de faire communiquer le serveur avec le client hébergé dans un véhicule sans processus complexe de déploiement, uniquement en utilisant une liaison TCP. Avec ce système des applications respectant l'identité visuelle du constructeur peuvent être rapidement développées.

- Le navigateur spécifique : Un autre outil de développement que nous utilisons dans cette thèse est une version spécifique du moteur de rendu HTML: webkit. Les navigateurs internet ont été développés spécifiquement pour les applications connectées. Ils supportent le protocole HTTP et peuvent rendre graphiquement des graphiques vectoriels et bitmaps. La version que nous utilisons est capable de s'interfacer avec un module de lecture des trames évènementielles envoyées sur le réseau CAN par un iDrive.

## 2.2 LA RÉVOLUTION NUMÉRIQUE: LES MÉTADONNÉES ET LA RECOMMANDATION

Les média numériques ne permettent pas uniquement d'améliorer la qualité du contenu, ou de faciliter sa duplication ou sa distribution que leurs équivalents numériques. Leur arrivée a en fait permis le développement d'un nouveau type d'information : les métadonnées. Ce chapitre explique pourquoi et comment les métadonnées constituent une contribution à la consommation de médias à la demande. Il analyse les différents types de métadonnées qui sont associées au contenu multimédia et le manière d'y accéder et de les utiliser dans un système embarqué. Ce chapitre illustre les différents types de flux de création, d'extraction, de dissémination et d'extraction de ce type d'information suivant les différentes entités de la chaîne de valeur du multimédia. Enfin, nous abordons le principal intérêt fonctionnel des métadonnées pour notre problème : la recommandation.

### *Des standards pour la représentation de la connaissance*

L'expression d'informations à propos de données multimédia a été normalisée et standardisée à de nombreuses reprises dans l'histoire de l'informatique. Les métainformations constituent une extension du contenu multimédia et peuvent être traitées en même temps que lui (*e.g.* afficher le titre d'un morceau de musique et le nom de l'artiste pendant qu'on le joue) ou de manière séparée (*e.g.* filtrer tous les titres d'un catalogue en fonction de leur année de parution). Il y a différents types de métadonnées: les métadonnées descriptives, les métadonnées administratives, les métadonnées de gestion des droits ou encore les métadonnées structurelles. Différents types de standards ont été mis au point : Certains comme Dublin Core, SMTPE ou encore EBU Core visent un objectif fonctionnel très précis ; respectivement, décrire un média avec uniquement 15 types de métadonnées, donner des labels à des trames vidéo ou un vocabulaire de description des programmes TV et radio ; Certains proposent de véritables structures polyvalentes comme le MPEG-7 et MPEG-21.

- Les données textuelles : Les données textuelles constituent la plus atomistique des représentations des métainformations. Le format du texte représente l'avantage d'être à la fois compréhensible par une personne et traitable par une machine. Une donnée textuelle peut à la fois représenter directement une information elle aussi textuelle comme le titre d'une œuvre ou représenter une donnée comme la date de sa production ou les données GPS d'une prise d'image. La norme MIME donne un premier niveau de standardisation des données textuelles de métainformation.
- Les données structurées : Le langage XML donne une structure aux données textuelles qui sont entourées de balises auxquelles on peut donner des attributs. La plupart du temps le format XML est stocké au format texte pour qu'un opérateur humain puisse le lire et l'éditer. Cependant, ce format de stockage représente certaines limitations en termes d'échange et de traitement. Le texte doit être analysé et converti en objets lorsqu'il doit être traité, c'est la sérialisation ; il est également sous-optimal de stocker des données comme des nombres flottants sous forme de texte. Une solution est de compresser les données avec un codeur entropique comme le format ZIP, mais cela ne ferait que rajouter une étape supplémentaire sans résoudre le problème du traitement. Une autre solution est d'utiliser un format binaire pour XML, qui permettrait un traitement rapide et un stockage plus efficace. Malheureusement la standardisation n'a pas encore abouti à l'adoption d'un format commun.
- Les données sémantiques : La quantité croissante de métadonnées en ligne sous forme standardisée a fait apparaitre deux tendances : D'un côté, on peut relier les données entre elles en utilisant des identifiants uniques pour localiser les ressources (respectivement les URI et les URLs) ; par ailleurs on peut exprimer des relations logiques entre ces données.

### *Décrire des objets multimédia*

Nous pouvons désormais représenter les métadonnées comme des objets informatiques. L'espace d'utilisation de ces métadonnées est infini c'est pourquoi nous allons nous focaliser sur le cas d'utilisation de notre scénario introductif.

### *La description des informations*

Dans la description des informations, on distingue principalement le thème du format. Le thème englobe les sujets principaux qui sont traités (par exemple, le sport, l'actualité politique ou économique, *etc.*). Le format définit la manière dont le programme est produit et/ou diffusé (on distingue ainsi un reportage, d'un flash info ou d'un talk-show). Aujourd'hui les podcasts et les vodcasts constituent le principal type d'information à la demande et utilisent une taxonomie définie par iTunes listant les principaux sujets d'information. Elle est composée de 17 catégories et de 51 sous catégories. Il s'agit donc de métadonnées structurées, mais non sémantiques. Des vocabulaires plus complexes sont actuellement conçus comme pour la sémantisation du NewsML proposé par Troncy [2010].

*La description du contenu musical*

Lorsque des auteurs comme Pachet and Cazaly [2000] and Aucouturier and Pachet [2003] se sont penchés sur les taxonomies musicales, ils ont commencés par regarder celles en usage dans la vente et sont arrivés à la conclusion qu'il n'y avait pas de manière unifiée de classer la musique. Les disquaires veulent vendre les CD entreposés dans leurs magasins et utilisent de ce fait un fléchage relativement simple à maintenir dans les rayonnages. Les discaires en ligne s'attendent à ce que leurs clients naviguent sur leur site par des systèmes d'onglets et de menus et ont donc développé des classifications plus complexes. Pour notre étude nous avons distingué les classifications suivantes :

- Informations concernant la présentation/représentation de l'œuvre : Tout le monde connait le nom de son artiste préféré et même celui des albums et des chansons qui l'on fait connaître. On peut ajouter à ses informations le type de performance (enregistrement studio ou concert), l'époque.
- Le genre musical : Le genre est un concept dit de 'haut niveau' qui permet de regrouper des œuvres musicales ayant des similarités dans leur composition. La notion de genre est certainement l'une des plus contestées, car la même dénomination (par exemple 'hip-hop') peur revêtir des significations complètement différentes dans des contextes culturels et géographiques différents.
- L'atmosphère : L'atmosphère d'une musique relate l'impression qu'elle provoque chez qui l'écoute (angoissante, entrainante, joyeuse, *etc.*). Une double échelle introduite par Russell [1980] permet de la définir à l'aide de deux données ; d'une part l'excitation (arousal) et d'autre par sa valeur ou intensité (valence).
- Informations rythmiques ou mélodiques : On peut décrire une musique à l'aide d'autre types de métadonnées comme le tempo, la tonalité, les instruments ou la présence de voix.

*La description des utilisateurs et des usages*

Utiliser des métadonnées ne se limite pas au contenu, on peut également les utiliser pour décrire les préférences des utilisateurs et leurs usages.

- Les préférences des utilisateurs : On peut associer aux différentes descriptions de contenu musical une valeur de préférence, par exemple comprise entre -100 et 100. Le standard MPEG-7 comporte par exemple un `UsagePreference` composé de `Browsing preferences` (préférences de sélection) ainsi que de `Filtering and search preferences` (préférences de filtrage et de recherche).
- L'historique des utilisateurs : On peut aussi pour chaque donnée multimédia (piste de musique, information, *etc.*) enregistrer

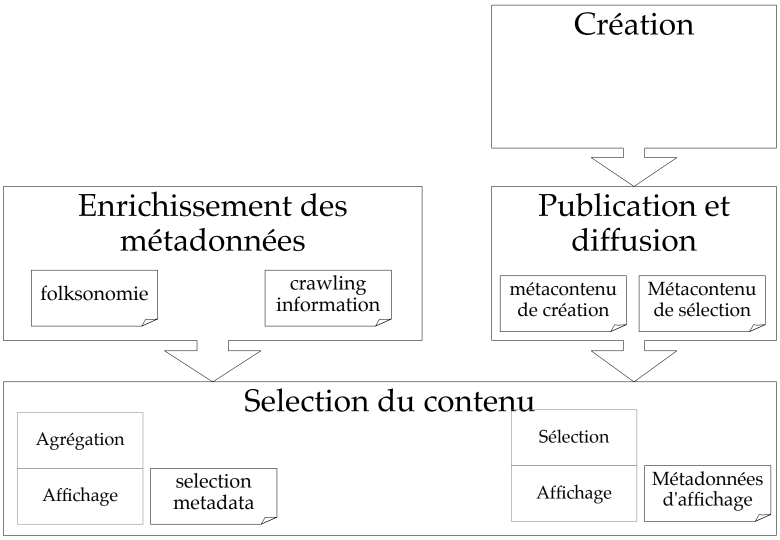


Figure 1: Les métadonnées dans la chaîne de valeur de distribution des médias (co-publiée dans Hahn *et al.* [2010])

l'usage de l'utilisateur, c'est à dire les évènements, dates et durées d'usage.

- Les données idiosyncrasiques : On peut associer des balises descriptives de l'usage du contenu (musique pour la relaxation, pour le sport, *etc.*).

*Les flux de traitement des métadonnées*

Dans l'ensemble de la chaîne de valeur de la production et la consommation de médias, les métadonnées sont produites par différentes entités et cela a des conséquences sur notre architecture. La figure 1, résume les différentes étapes d'enrichissement des métainformations de la chaîne de valeur de la distribution de musique.

EXTRACTION ET CRÉATION On distingue deux paradigmes dans le genèse de la métainformation : (1) L'extraction de métadonnées à partir du contenu multimédia lui-même par des méthodes algorithmiques (méthode dite de 'bas-en-haut') et (2) la création de métadonnées par des données exogènes (méthode dite de 'haut-en-bas'). L'extraction de métadonnées peut être effectuée localement et dépend principalement des algorithmes de traitement du signal. La création de métadonnées peut se baser sur différentes sources comme les données éditoriales, les avis d'experts, les folksonomies issues des réseaux sociaux ou encore les études sociologiques. Suivant la manière dont elles sont extraites, les métadonnées sont délivrées en accompagnement ou séparément du contenu, ce qui peut engendrer une charge réseau supplémentaire. Par ailleurs, lorsque des métadonnées sont agrégées depuis des sources multiples, il est nécessaire de pouvoir fusionner les informations. Des techniques de comparaison de chaînes de caractères comme la distance de Levenshtein ou le Soundex peuvent être utilisées, car elles sont plus efficaces que la distance de Hamming.

LE CAS SPÉCIFIQUE DE LA RECOMMANDATION La recommandation de contenu est un exemple très illustratif de la diversité des usages potentiels des métadonnées. La recommandation consiste à suggérer à un utilisateur du contenu en se basant sur différents types d'informations. On distingue la recommandation basée sur le contenu, de la recommandation sociale. Pour le contenu musical, la recommandation peut se baser sur des données acoustiques afin d'établir une similarité entre morceaux de musiques ou entre artistes. Il est également possible d'utiliser des métadonnées issues d'annotations d'experts pour établir cette similarité comme c'est le cas chez Pandora avec le *Music Genome Project*. Dans le filtrage collaboratif, on se base sur les préférences d'un ensemble d'utilisateurs sur un ensemble d'items d'un catalogue comme des morceaux de musique pour prévoir les items sélectionnés par les utilisateurs. Le filtrage collaboratif peut-être centré sur les items, *i.e.* deux items sont considérés comme similaires si les utilisateurs ont tendance à leur donner la même valeur de préférence ; ou bien centré sur les utilisateurs, *i.e.* deux utilisateurs sont considérés comme similaires si à contenu identique, ils donnent les mêmes valeurs de préférence. Cette dernière méthode ne semble pas donner les meilleurs résultats. Les approches combinant les deux types d'informations ont tendance à donner les meilleurs résultats.

### *Conclusion*

Les métadonnées constituent une révolution en ce qu'elles apportent du sens au contenu multimédia et l'accompagnent dans les différentes étapes de sa vie, de la création jusqu'à l'utilisation. Ce 'sens' ne se limite pas à des définitions abstraites, mais se concrétise par la création de données sémantiques de hauts niveaux utilisables dans une programmation logique pour trier, filtrer, annoter, adapter des morceaux de musique, ou des vidéos. Le contenu multimédia ne se limite donc plus à des signaux visuels et auditifs numérisés que l'on peut coder et décoder, mais à des objets dotés de propriétés propres et génériques. La multiplicité des sources de métadonnées doit être prise en compte lors de la conception d'un système de consommation d'infodivertissement ubiquitaire, nous y reviendrons dans les prochains chapitres.

## 2.3 LA FOURNITURE DE CONTENU À LA DEMANDE, ARCHITECTURE DU SYSTÈME PERSONAL RADIO

La consommation mobile de médias peut se découper principalement en deux paradigmes : (1) d'un côté on distingue les flux poussés dans lequel le média est envoyé de manière synchrone au récepteur et (2) d'un autre côté les flux tirés qui regroupent les média dits 'à la demande' dans lesquels les utilisateurs sélectionnent et téléchargent des médias qu'ils ont choisis au moment où ils souhaitent les consommer.

### *Les différentes architectures de distribution des médias*

- L'architecture en diffusion : Elle consiste à envoyer un signal d'une antenne terrestre ou satellite à un nombre inconnu de récepteurs. L'architecture en diffusion est particulièrement bien adaptée aux flux poussés et s'adapte très bien à un grand nombre d'utilisateurs puisque l'ajout d'un récepteur n'a pas d'influence sur la performance de l'ensemble de l'architecture. Toutefois, dans cette architecture, le processus de sélection est complètement laissé au client, le réseau se chargeant de la distribution.

- L'architecture routée : Les architectures routées de type internet sont constituées de serveurs web qui contiennent des données multimédias connectées au réseau. Les clients eux-mêmes connectés sont capables de communiquer de manière bidirectionnelle avec les serveurs. Le client interroge le serveur sur les ressources disponibles (comme le client développé par Rhapsody) et l'utilisateur peut sélectionner parmi les choix qui lui sont proposés (*e.g* sélectionner les musiques du catalogue Rhapsody qui lui convienne) . La communication entre le client et le serveur autorise une consommation synchrone ou asynchrone des ressources, une adaptation des ressources (*e.g.* en fonction des capacités du client, ou de l'état du réseau, *etc.*) et enfin l'implantation de fonctions de retour de pertinence.

- Les architectures pair-à-pair : Ces architectures sont basées sur une surcouche de type réseau au niveau de la couche application du modèle OSI. Elles sont basées sur l'idée que chaque pair se comporte à la fois comme un serveur et un client. L'index des ressources partagées entre les pairs peut être centralisé (comme implanté par le système Spotify) ce qui permet une recherche centralisée ou au contraire partagé en utilisant des tables de hachage distribuées.

Une étude des avantages et inconvénients des différents types d'architectures de distribution de média nous a conduits à choisir une architecture routée avec un client riche implanté pour la voiture et une architecture multicouche de serveurs pour les services d'infodivertissement personnalisés. En effet cette architecture permet de proposer des flux tirés (à la demande), mais reste compatible avec des flux poussés en diffusion. Un grand nombre de services internet proposent déjà des interfaces à la demande et nous allons à présent traiter de leur intégration dans un service pour automobilistes.

### *Personal Radio : Une architecture pour une distribution de services personnalisés*

#### *Le client*

L'architecture fonctionnelle du client comprend les éléments suivants:

- Entrée de l'utilisateur : L'utilisateur doit pouvoir interagir avec l'application et donner différents types d'entrées. Recherche en texte, sélection de listes, demande d'informations supplémentaires sur un titre, génération automatique d'une liste.

- Résolution et accès des médias : Quand un média est sélectionné le client doit savoir s'il est déjà présent localement ou s'il doit le télécharger. Par ailleurs, les dispositifs de DRM doivent permettre à différents types de modèles d'affaires de cohabiter.

- Stockage des médias et indexation : Afin de parer à de interruptions intempestives de la connexion au réseau mobile qui sont susceptibles de se produire lorsque la voiture roule, le client doit télécharger et stocker les médias sélectionner afin de les mettre à disposition de l'utilisateur lorsque celui-ci est hors-ligne

- Affichage et rendu des médias : Le rendu des médias audio consiste principalement à décoder le signal audio compressé. Dans notre première version du prototype, nous n'avons pas proposé de fonctions telles que la mise en pause, l'avance et le retour rapides, car nous les avons jugé peu intéressants et dangereux pour un usage dans un véhicule.

L'implantation du client du prototype de *Personal Radio* comprend les différents éléments techniques suivants :

- Interface utilisateur : Elle est basée sur le dispositif Remote HMI présenté plus haut. La logique de l'application est une bibliothèque chargée par le serveur Remote HMI qui tourne sur un PC Linux embarqué dans la voiture. L'affichage sur l'écran CID est effectué par le CIC-High qui héberge un système d'exploitation QNIX capable de gérer les interactions avec les différents boutons et molettes à disposition dans la console centrale du véhicule.
- Profiler : Chaque utilisateur dispose d'un profil stocké sur un serveur détaillé plus loin. Le profil contient les listes de lectures personnalisées et son historique de consommation. Le profil est synchronisé entre le client et le serveur lorsque le premier se connecte au second.
- L'intergiciel : il consiste en la couche logicielle entre le véhicule et les services en ligne avec principalement deux rôles ; (1) l'échange de métadonnées avec le serveur terminal BMW et (2) le téléchargement de média directement auprès des serveurs des fournisseurs de contenu.

Nous avons implanté deux prototypes, un premier pour le développement émule les différents composants d'un véhicule sur un PC Windows et permet de développer l'application sans avoir à la déployer dans un véhicule pour la tester. Un deuxième utilise les composants natifs du système Remote HMI dans une BMW série 7 (modèles F01 et F02).

### *Les services en ligne*

Par ailleurs, nous avons opté pour une architecture multicouche pour la fourniture de services.

LE SERVEUR TERMINAL Son rôle est (1) d'abstraire les services des différents services de musique et podcast en ligne. En effet il serait trop complexe de développer une interface par fournisseur de contenu, car au moindre changement, il faudrait mettre à jour le logiciel hébergé dans le véhicule. Or nous avons vu plus haut, que le monde des services en ligne est caractérisé par une haute volatilité. Nous avons donc développé une interface unique entre le véhicule et le serveur BMW. Par ailleurs, le serveur terminal permet (2) d'agréger des données en provenance de plusieurs services pour générer des listes de lecture mixant par exemple musique et information. Enfin (3) le serveur permet la persistance des informations des profils utilisateurs en pilotant une

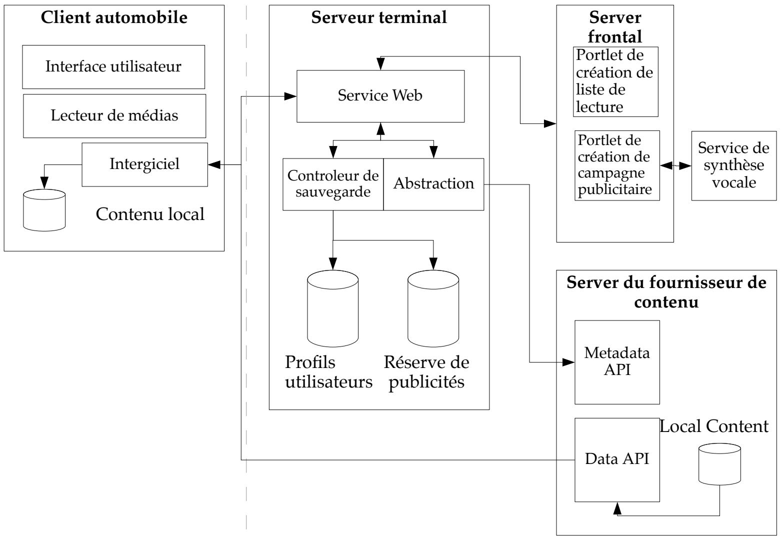


Figure 2: Intégration de l'outil de définition de campagnes personnalisées dans *Personal Radio*

base de données. Le service web pour les véhicules ainsi que les interfaces avec les fournisseurs de contenu sont implantés sous forme de servlets java et tournent sur un serveur Glassfish. Les données sont stockées dans une base de données relationnelle MySQL et transformées en objets par une couche d'abstraction basée sur Hybernate.

LE SERVEUR FRONTAL Il s'agit d'une interface web qui peut être utilisée avec un navigateur internet. Il est basé sur un moteur de portlets (liferay). Il contient deux types de portlets : (1) des portlets d'administration pour gérer des listes de lectures fournies par défaut aux utilisateurs et (2) la communication avec le système de persistance de la base de données.

Pour illustrer l'intérêt et la flexibilité de cette architecture à plusieurs niveaux, nous avons implanté une application de création de campagnes publicitaires audio pour les clients BMW (voir aussi illustration 2). En nous appuyant sur les fonctionnalités d'agrégations de média, nous avons pu élargir les fonctionnalités de notre prototype en nous appuyant sur les éléments suivants :

- Configurabilité en ligne : En nous appuyant sur le serveur frontal, nous avons créé une portlet pour définir une publicité (titre, image affichée sur l'écran, texte).
- Agrégation : L'architecture *Personal Radio* est capable de générer une liste de lecture mélangeant musique, information et publicité. Nous avons connecté le serveur terminal à une application de synthèse audio, pour pouvoir générer à la demande un texte audio décrivant le contenu de l'annonce publicitaire.
- Personnalisation : La synthèse audio nous permettait de personnaliser l'annonce publicitaire au nom du client et de sélectionner les publicités adéquates en fonction de son profil d'acheteur de produits BMW.

- Abstraction : Les listes de lecture étant génériques pour le véhicule, aucune modification du logiciel du véhicule n'était nécessaire.

### *Conclusion*

Les architectures web de type client/serveur sont particulièrement bien adaptées à la distribution de contenu à la demande. Pour contourner le problème de segmentation des différents fournisseurs de contenu qui proposent des interfaces non standardisées à leurs services, nous avons dû introduire une couche d'abstraction sur un serveur distant qui nous permet de minimiser la complexité du client embarqué dans le véhicule. La flexibilité de notre architecture est illustrée au travers d'une extension des services de *Personal Radio* par une fonctionnalité de diffusion de publicités personnalisées à travers les listes de lecture multimédia. Toutefois, le plein potentiel des contenus à la demande nécessite une utilisation plus fine des possibilités de personnalisation que nous allons étudier dans les prochains chapitres.

## 2.4 LA PERSONNALISATION INTERACTIVE DU CONTENU

Après avoir introduit un système de fourniture à la demande, nous devons nous pencher sur la concrète de l'expérience utilisateur. Ce chapitre donne une solution à la personnalisation lorsqu'elle est prise comme un problème d'interaction entre le véhicule et le conducteur. Nous nous focalisons en particulier sur la musique puisque c'est le média le plus consommé dans les véhicules. Nous analysons l'ergonomie des fonctionnalités de recherche et de sélection de médias dans un véhicule. Enfin nous décrivons la conception d'un navigateur musical spécifiquement pour un usage automobile et qui s'intègre à la plateforme mobile que nous venons de décrire.

### *Position du problème*

Fournir du contenu à la demande libère l'utilisateur de la programmation des radios et des télévisions qui développent chez les utilisateurs le phénomène de 'zapping' pour naviguer entre 'hits'. Toutefois, un problème de cette nouvelle liberté expliquée par Anderson [2008] est qu'entre choisir des artistes très populaires ou des nouveautés encore inconnues, les utilisateurs doivent pouvoir s'y retrouver. Ainsi la longue traine devrait faire apparaître des modèles d'affaires non plus basés sur la vente massive d'un seul artiste, mais sur de faibles volumes sur une grande diversité. C'est le paradoxe du choix de Schwartz [2005], la diversité peut-être frustrante si elle n'est pas accompagnée d'un effort de classification pour aider l'utilisateur et c'est pourquoi nous devons adapter l'interface utilisateur de notre solution automobile en conséquence. Nous allons ici proposer des solutions aux problèmes suivants :

- Faire une interface graphique compatible avec les modalités du véhicule
- Agréger l'information des fournisseurs de service pour permettre la recommandation

- Développer un architecture technique extensible qui permette d'atteindre nos objectifs fonctionnels

*Parcourir et rechercher de la musique dans un véhicule ou rendre l'utilisateur actif sans le distraire*

La recherche de musique et la création de listes de lecture représentent une charge cognitive potentielle pour le conducteur du véhicule qui est supérieure au simple allumage d'un autoradio.

*Fondamentaux psychologiques et cognitifs*

D'un côté la musique est un média divertissant et les études de Sloboda *et al.* [2001] et Dibben and Williamson [2007] soulignent qu'en écouter peut améliorer l'attention du conducteur, sa réactivité et sa concentration. En particulier, dans les situations d'ennui (route longue et monotone) ou de stress (embouteillages), la musique est un moyen d'améliorer l'environnement de l'utilisateur. Cependant, les systèmes connectés automobiles sont une source majeure de distraction comme le fait remarquer Lansdown [2000] en raison des temps de chargement imprévisibles et des informations dynamiques parfois incohérentes. Parmi les trois classes de tâches effectuées par un conducteur de voiture spécifiées par Geiser [1985], les tâches liées à l'infodivertissement appartiennent aux moins prioritaires (ce sont des tâches tertiaires dites de confort) et ne doivent pas interférer avec les tâches primaires et secondaires qui concernent le contrôle du véhicule, la prise d'information ou la signalisation des changements de direction.

Non seulement les systèmes connectés sont des sources de distraction par eux-mêmes, car ils engendrent des tâches supplémentaires pour le conducteur, mais ils sont une deuxième source d'insécurité en ce qu'ils peuvent conduire à une frustration de l'utilisateur s'ils sont mal conçus : surcharge visuelle si les informations sont mal présentées, surcharge biomécanique si les interfaces haptiques sont inadéquates, surcharge auditive si le signal audio masque le bruit du moteur ou de la rue et empêche la prise d'information.

Un certain nombre de règles on été listées par une métaétude de Lansdown [2000], nous nous focaliserons sur les suivantes dans la conception de notre système :

- prise dans sa globalité, toute tâche ne devrait pas durer plus de 15 secondes [Green, 1999]
- la durée moyenne d'un coup d'œil doit être de moins de 1.2 seconde [Dingus *et al.*, 1989]
- aucun coup d'œil ne devrait durer plus de deux secondes [Zwahlen *et al.*, 1988] et [Zwahlen and DeBald, 1986]

L'utilisation des modalités doit être optimisé [Sarter, 2006] pour minimiser la charge cognitive :

MODALITÉ VISUELLE C'est certainement la modalité la plus riche en termes d'informations qui peuvent être transmises à l'utilisateur :

- Informations textuelles : les navigateurs musicaux pour stations de travail proposent tous des recherches plein texte sur les titres et les artistes qui sont affichés à l'écran. La pertinence d'utiliser d'autres types d'information pour décrire une musique varie selon leur catégorie. Les émotions et le contexte d'utilisation semblent être les plus parlants. Lors du parcours de listes textuelles, l'utilisation d'effet zoomés de type 'fisheye' permet d'informer l'utilisateur des items bénéficiant du focus applicatif.

- Images (informations picturales) : Même si elles n'ont pas la précision du texte, les images peuvent permettre d'illustrer rapidement une information comme l'album à laquelle appartient la musique actuellement jouée.
- Informations topologiques : Lors de la représentation d'un catalogue, on peut utiliser des informations topologiques visuelles pour illustrer la densité du catalogue (nombre de morceaux de musique d'un certain genre par exemple) ou la proximité entre différents éléments du catalogue. La notion de navigation dans le catalogue peut alors se rapprocher du déplacement sur une cartographie, tâche déjà éprouvée pour l'aide à la navigation avec les systèmes GPS automobiles [Rölle, 2007].

MODALITÉ AUDITIVE Pour la musique la modalité auditive joue évidemment le rôle crucial du rendu final du média, mais cette modalité peut également être utilisée comme dispositif d'entrée utilisateur.

- Menus avec signalétique sonore : Lorsqu'il parcourt les différents onglets d'un menu, il est possible de proposer à l'utilisateur une signalétique sonore afin de l'aider à savoir où il se positionne sans avoir à regarder l'écran. La synthèse vocale, ainsi que les spearcons et autres spindex (basés sur des contractions syllabiques de mots) ont été testés par Jeon *et al.* [2009] pour le monde automobile avec un certain succès.
- Reconnaissance vocale et recherche mélodique : Bien qu'introduite dans les téléphones portables depuis longtemps, la reconnaissance vocale attend encore d'être intégré au monde automobile. De fortes réserves quant aux limitations cognitives sont pour l'instant avancées par [Lee *et al.*, 2001] et Strayer *et al.* [2003] à l'idée de faire parler une personne qui conduit. La recherche mélodique est également une possibilité plus adaptée à la musique, mais la quantité de métadonnées nécessaires pose des problèmes de ressources en termes de puissance de calcul et de réseau encore difficilement surmontable dans un véhicule, sans sacrifier à la réactivité.

MODALITÉ HAPTIQUE Cette modalité est presque exclusivement utilisée pour les entrées de l'utilisateur.

- Entrées par saisie ou sélection : La saisie de données textuelles ou la sélection de menus peut être effectuée soit par l'utilisation de boutons, molettes et d'écran tactile. Ces derniers ne font pour l'instant pas l'objet d'un consensus dans le monde automobile à cause de l'importante charge visuelle qu'ils nécessitent. Les méthodes prédictives de type T9 permettent de diminuer le nombre d'interactions nécessaires à la saisie de texte en appuyant sur un nombre limité de boutons.

- Autres types d'entrées : Geiger [2003] a étudié dans sa thèse un mode de reconnaissance de mouvement de la main par des capteurs infrarouges pour utiliser des fonctionnalités d'infodivertissement comme la navigation au travers de listes. Une autre manière de rechercher de la musique est de taper le rythme du morceau recherché. C'est une méthode analogue à la recherche mélodique et a déjà été implantée dans des services en ligne.

*Personnalisation active : les tâches de l'utilisateur*

- La recherche : La formulation d'une requête est un compromis entre le temps pour formuler la requête et sa granularité. Plus on crée un filtre spécifique, meilleures sont les chances de trouver ce que l'on cherche, mais plus la requête doit être affinée et plus la charge cognitive est importante. On distingue principalement la recherche plein texte très adaptée à une recherche sur un corpus limité d'items, de la recherche par catégories bien adaptée à la recherche sur de très larges corpus. La sélection des résultats peut se faire suivant une liste sous forme de texte, dans notre prototype nous combinons à cette liste les images des albums pour une identification plus rapide.

- La création de listes de lectures : Il s'agit de l'étape ultime dans l'expérience utilisateur. Dans la première version de *Personal Radio*, la création d'une liste de lecture était basée sur la similarité par rapport à un seul morceau de musique. Nous allons proposer un système permettant de définir rapidement des filtres multicritères.

*L'intégration de la recommandation dans un système mobile*

Avant d'étudier la solution que nous proposons pour le monde automobile, il est nécessaire d'étudier l'état de l'art des systèmes interactifs pour la navigation musicale. Nous nous apercevons en réalité que les approches sont diverses et mettent en avant des techniques que nous nous proposerons de combiner, voire d'étendre pour l'usage automobile.

UN NAVIGATEUR LOCAL AVEC AGRÉGATEUR DE MÉTADONNÉES CONNECTÉ La MusicBox développée par Lillie [2008] aux MIT Media Labs propose une solution pour poste de travail basée sur l'analyse du contenu localement présent sur l'ordinateur de l'utilisateur. Elle consiste à effectuer une extraction de métadonnées à l'aide de services en ligne (extraction de métadonnées mélodiques, de balises de réseaux sociaux, *etc.*) et à synthétiser ces informations à l'aide de techniques de fouille de données (analyse en composante principale). Le catalogue peut alors être représenté en deux dimensions par un nuage de points, projeté sur les axes les plus discriminants. La création de liste lecture se base par une sélection à la souris d'une zone de cette 'cartographie'. La MusicBox illustre bien la notion d'un composant d'agrégation d'information, mais son intégration dans le client est un choix que nous ne pouvons transposer au monde automobile. D'une part, il est impossible d'avoir accès à l'ensemble des données audio du catalogue en ligne, par ailleurs, le mode de sélection proposé demande une charge visuelle bien trop importante.

RECOMMANDATION MUSICALE EN LIGNE Un système de recommandation en ligne comme MusicLens a introduit un concept important dans la formulation d'une requête : la prévisualisation. MusicLens[3] propose à l'utilisateur de déplacer des curseurs pour régler la composition de la liste de lecture avec des notions comme le tempo, la présence de voix, l'usage ou encore l'époque. Avant même que la liste de lecture ne soit générée, la page web présente une vision en 3D de la requête. Ce n'est pas à proprement parler une prévisualisation, mais cela permet à l'utilisateur de se faire une idée rapide de l'influence des paramètres qu'il a modifiés. Dans un système automobile où le retour visuel doit être rapide, il faut pouvoir s'assurer qu'une prévisualisation de la requête prend un temps raisonnable. Cela peut-être un problème lorsque l'on agrège des données de plusieurs fournisseurs de contenu qui ont des temps de latence différents.

SYSTÈME MOBILE AVEC DÉFINITION SÉMANTIQUE DES PRÉFÉRENCES Le projet Mobi Xpl radio est un prototype de découverte de radios musicales pour téléphone portable [Noppens *et al.*, 2007b,a]. Il se base sur une architecture en plusieurs couches avec un client très léger qui permet la définition sémantique de préférence. L'utilisateur peut formuler des assertions logiques complexes comme "je préfère le genre B au genre A". La combinaison de catégories pour la définition de listes de lecture semble en effet bien se combiner à la diversité des informations disponibles sur internet. L'architecture proposée par MobiXpl est une solution intéressante particulièrement bien adaptée aux choix faits dans le chapitre précédent.

### *Notre prototype de navigateur musical pour automobiles*

Nous avons implanté un prototype de navigateur musical fort des analyses que nous venons de décrire[4]

#### *Les exigences fonctionnelles et non fonctionnelles*

Le navigateur doit pouvoir permettre les fonctionnalités suivantes :

- Création de listes de lectures musicales basées sur un filtre composé de plusieurs critères ;
- Découverte facile et aisée des différentes options de création des filtres ;
- Prévisualisation rapide des résultats de la requête avant la création complète de la liste de lecture ;
- Possibilité pour l'utilisateur de sélectionner des résultats alternatifs sans avoir à recréer un nouveau filtre.

Par ailleurs, nous avons pour objectif d'utiliser un catalogue en ligne et d'utiliser différents services en ligne comme source de métadonnées ainsi que les réseaux sociaux pour permettre une sauvegarde et une utilisation étendue des préférences musicales de l'utilisateur.

---

[3] www.musiclens.com

[4] La réalisation elle-même du prototype a été réalisée en collaboration avec deux étudiants en projets de fin d'études : Clemens Hahn et Sascha Gebhardt

*Les composants logiques issus de Personal Radio*

Nous reprenons en les étendant à nos besoins les composants suivants du chapitre précédent :

LE SERVEUR TERMINAL Il réalise l'interface d'abstraction en présentant au véhicule un ensemble unifié de schémas de communications pour la recherche de contenu par critères multiples, l'obtention de métadonnées supplémentaires comme les images ou des morceaux similaires.

LE FOURNISSEUR DE CONTENU MUSICAL Nous avons utilisé comme dans le chapitre précédent un fournisseur de contenu musical que nous pouvons interchanger avec n'importe quel autre en utilisant la couche d'abstraction suscitée.

*Les composants spécifiques au navigateur musical*

LE CLIENT Nous avons redéveloppé un client avec une interface deux parties. Un premier écran de présentation des différents filtres dans lequel l'utilisateur peut ajouter, modifier ou supprimer des filtres. Il y a quatre sortes de filtres : le genre, l'ambiance, la popularité, l'époque et la source. Nous avons adapté la présentation et la sélection des filtres à l'usage avec la molette de l'iDrive.

L'AGRÉGATEUR DE MÉTADONNÉES Nous avons du ajouter au serveur terminal un composant d'agrégation de métadonnées afin de pouvoir fusionner les métadonnées en provenance de différents services. Ce composant est capable de comparer des chaînes de caractères afin de contourner les différentes orthographes pouvant être utilisées pour le même artiste ou le même titre. Des métadonnées sont ainsi agrégées depuis les services de The Echonest pour la popularité des morceaux, Last.fm pour les réseaux sociaux (et l'utilisation fonctions de similarités sociales), Rhapsody pour les différents genres ainsi que Gracenote pour l'ambiance musicale.

LE PRÉCHARGEMENT ET LA PERSISTANCE DE MÉTADONNÉES Afin de permettre un mécanisme de prévisualisation des résultats d'une requête de liste de lecture, les métadonnées d'un nombre significatif de titres sont préchargées par le serveur terminal et sauvegardées dans une base de données relationnelle. Ceci permet au générateur de liste de lecture de filtrer sur sa propre base un certain nombre de titres qui sont retournés immédiatement en réponse au véhicule. Ainsi, les fournisseurs de contenus ne sont pas utilisés pour la prévisualisation des résultats de la requête ce qui permet de réduire significativement la latence du système. Par ailleurs, le client peut commencer à télécharger les premiers titres dès qu'il reçoit les résultats de prévisualisation.

LE GÉNÉRATEUR DE LISTE DE LECTURE Le générateur de liste de lecture se base sur l'agrégateur de métadonnées pour étendre les premiers résultats issus du filtrage local des métadonnées dit titres générateurs. Pour chaque titre générateur, des titres similaires sont agrégés depuis les fournisseurs de métadonnées. Les titres sont ensuite ordonnés en une chaîne suivant le nombre de titres similaires qu'ils ont afin de les regrouper par proximité, les titres agrégés qui leur sont similaires sont placés entre eux puis sont intercalés les titres restants. Ceci permet d'assurer à la liste de lecture un tonalité musicale homogène tout en utilisant des sources hétérogènes.

### *Implantation technique*

Le client a été développé en Flash Action Script 3 et tourne dans un navigateur internet spécifique basé sur le moteur HTML Webkit et il peut être relié à une interface CAN pour traduire les trames envoyées par la molette de l'iDrive en évènements DOM qui sont renvoyés à l'applet flash par un mécanisme JavaScript. Le navigateur tourne dans un PC Windows embarqué dans le coffre d'une BMW série 7 dont la sortie DVI et transformée en LVDS pour l'écran CID. Un routeur UMTS permet de connecter le client à internet. Le serveur terminal tourne sur la même infrastructure que pour *Personal Radio*, à savoir un serveur glassfish et une base de données MySQL.

### *Conclusion*

La diversité des médias disponibles via internet est un véritable enjeu pour le développement de services de divertissement réactifs, non intrusifs et procurant au conducteur un véritable gain de confort. En permettant d'accéder à la demande à du contenu, nous avons ouvert des possibilités de personnalisation pour lesquelles le conducteur doit être guidé, encadré et même conseillé afin de ne pas le dévier de sa tâche principale : la conduite. La personnalisation active nous a obligés à revoir les fonctionnalités du client embarqué, mais également à compléter les composants de l'architecture distante avec un mécanisme d'agrégation et de préchargement des métadonnées assurant une grande réactivité à l'ensemble.

## 2.5 LA PERSONNALISATION CONTEXTUELLE PASSIVE DU CONTENU

Nous l'avons vu dans les chapitres précédents, la personnalisation de l'expérience utilisateur pose des problèmes, car elle peut accroitre la charge cognitive nécessaire à l'utilisation d'un système d'infodivertissement interactif, elle peut également se baser sur des critères non centrés sur l'utilisateur comme l'environnement de conduite. Ce chapitre, présente le cadre applicatif Program Director qui est une solution prototype offrant des réponses aux problèmes de raisonnement contextuel, de persistance des préférences contextuelles et de dissémination de ces préférences entre différentes entités logiques.

### *Position du problème*

Comme l'ont suggéré les comportementalistes des médias, l'usage du contexte comme filtre pour adapter le contenu et créer automatiquement des listes de lecture est une solution qui doit être prise en considération pour simplifier l'usage de l'infodivertissement et permettre à l'utilisateur la découverte de nouveaux médias. En réalité, l'usage d'informations contextuelles dans les systèmes dits 'intelligents' est une idée qui a fait long-feu dans la recherche en informatique et en automatique. D'un côté, des chercheurs se sont penchés sur les architectures d'adaptation ou de sélection de contenu alors que d'autres ont étudié la problématique de la distribution d'informations contextuelles. Dans notre scénario, l'usage de critères de recherches pour définir une liste de lecture peut être fortement influencé par le contexte. On atteint cependant là un fossé important de la recherche actuelle en matière de moteurs d'indexation. D'un côté des algorithmes très puissants ont été développés afin d'améliorer la vitesse d'indexation, la rapidité des recherches et leur précision, mais trop souvent sur des sujets trop focalisés sur le contenu. Pour pouvoir contrôler ces puissants outils, il manque encore une connaissance systématique des besoins utilisateurs afin de pouvoir automatiser non pas la recherche du contenu, mais sa sélection. Lieberman [2009] définit les systèmes d'intelligence artificielle comme des agents capables de créer un lien entre un objectif et une action. Comme nous le verrons, certains essaient d'extraire cette intelligence à partir d'une heuristique statistique, ce que Lieberman considère comme insuffisant. Pour lui, le 'bon sens' est difficile à extraire de données purement quantitatives. Hélas, le bon sens est aussi difficile à formaliser d'un point de vue informatique.

### *Les systèmes sensibles au contexte*

L'une des premières études des systèmes sensibles au contexte a été effectuée par Schilit [1995] dans une thèse où il se focalise sur des agents capables de se reconfigurer en fonction d'informations contextuelles. Il y donne une définition générale des problèmes de résolution et de notification entre serveurs et clients, thématique reprise par Huber [2008] avec l'idée de 'producteurs' d'information contextuelle auxquels des 'consommateurs' peuvent s'abonner pour être informés de nouvelles informations. Les systèmes proposés dans ces études n'apportent cependant pas de solution à un problème crucial : la prise de décision, *i.e.* une fois en possession des informations contextuelles. C'est chez Zimmermann *et al.* [2005] que l'accent est mis au niveau architectural sur la mise en place d'une couche sémantique entre les capteurs de données contextuelles et une couche de contrôle des applications. Le rôle de cette couche est d'extraire à partir de données brutes des concepts de haut niveau qui puissent être réutilisés par un raisonneur.

### *Le raisonnement contextuel appliqué : un état de l'art*

La plupart des travaux suscités sont des réflexions générales sur l'architecture de systèmes sensibles au contexte. Toutefois, il reste nécessaire d'implanter une méthode de raisonnement à partir de données contextuelles. Comme nous allons le voir, le type de raisonnement dépend principalement de deux choses : le type de données contextuelles et le type d'action qui doit être entrepris par le système. Nous pouvons reprendre les différentes étapes de l'expérience d'infodivertissement telles que nous les avons définis pour identifier les types d'actions associées suivant la taxonomie de Dey and Abowd [2000] (*cf.* tableau 1).

### *Le raisonnement basé sur des règles*

Les règles sont formées d'une expression logique exprimée sous forme d'une syntaxe du type SI context_information EST type_de_contexte ALORS choisir Resource. Ces règles sont basées sur une connaissance *a*

| | Présentation d'informations | Éxecution automatique d'un service | Balisage d'information par des données contextuelles |
|---|---|---|---|
| Découverte | × | | |
| Formulation de la requête | | × | |
| Filtrage | × | × | |
| Sélection des résultats | × | | |
| Rendu | × | | |
| Modification | | × | × |

Table 1: Les types d'actions contextuelles dans un système d'infodivertissement d'après la taxonomie de Dey and Abowd [2000]

*priori* et doivent être prédéfinie par un expert, un utilisateur ou faisant partie du 'bon sens'.

- Contenu balisé par des informations contextuelles : Cela consiste à associer directement à chaque titre d'une bibliothèque des informations de consommation contextuelle ([e.g. Reddy and Masciaa, 2006]).
- Les ontologies d'utilisation : Les ontologies permettent de définir des ensembles de concepts de haut niveau pour décrire les relations logiques entre métadonnées afin de généraliser les règles d'adaptation.

*Le raisonnement basé sur des informations statistiques*

Au contraire, on peut considérer que les règles reliant les informations contextuelles à une décision peuvent être apprises automatiquement. Omojokun *et al.* [2008] a introduit l'idée d'un motif entre les propriétés du signal audio d'un morceau choisi par un utilisateur et son environnement en se basant sur une analyse statistique compilant des données aussi variées que la température, l'humidité, le taux de précipitation, le déplacement de l'utilisateur, niveau sonore du bruit en les mettant en regard de soixante-dix différents descripteurs du signal audio.

D'une manière similaire, le prototype XPod [Dornbush *et al.*, 2007] utilise un réseau de neurones pour prédire la probabilité que l'utilisateur interrompe un morceau pour passer à un autre, en fonction de données physiologiques et cinématiques.

Comme nous pouvons le voir, une approche purement statistique est donc complètement intégrée à l'action que l'on veut prédire, et génère des solutions locales et spécifiques qui peuvent être difficiles à généraliser.

### *Intégration des systèmes sensibles au contexte dans une architecture de distribution de médias*

L'intégration des systèmes sensibles au contexte dans une architecture de distribution de médias revêt dans notre scénario une importance particulière, car rappelons le, une voiture dispose d'une connectivité limitée et d'un hardware embarqué aux contraintes fortes en matière de possibilité de mise à jour et de puissance de calcul.

#### *Systèmes distribués et architectures composites*

Weiß [2009] suggère de répartir la sélection du contenu entre le client et le serveur : Dans un premier temps le serveur sélectionne le contenu en fonction du profil de l'utilisateur et ensuite le client refiltre cette présélection en fonction de données contextuelles. De ce fait, le profil global de personnalisation doit être réparti entre le client et le serveur suivant un protocole de synchronisation que Weiß ne précise pas.

#### *Architectures 'proxy'*

Plutôt que de modifier la structure même architectures de sélection du multimédia, afin de leur intégrer la prise en compte du contexte, il est possible de procéder par une approche composite, réutilisant l'existant en matière d'indexation et de distribution. C'est l'approche dite en 'proxy' prise par Keidl [2004] ou Kaltz [2006]. L'idée est de transformer les requêtes du client au niveau d'un serveur intermédiaire en prenant en compte des informations contextuelles.

### *Le Vehicle Program Director*

Á la lumière de notre analyse nous avons développé le *Vehicle Program Director*, une solution prototype dont l'objectif est d'apporter les réponses suivantes au problème de personnalisation passive du contenu :

- Réutilisation de l'architecture *Personal Radio* existante
- Prise en considération de tout type d'informations contextuelles intérieures et extérieures au véhicule
- Implanter un mécanisme de persistance des préférences utilisateur en matière de sélection contextuelle
- Proposer une interface de définition par un expert ou un utilisateur des préférences et permettre un mécanisme d'apprentissage de ces préférences

#### *Les informations contextuelles du véhicule*

Nous avons donc commencé par construire une ontologie de différents types de contexte du véhicule.

CONTEXTE PROPRE AU VÉHICULE Ce contexte comprend l'environnement du véhicule : sa cinématique (vitesse, accélération,*etc.*), certaines informations météo (température extérieure, précipitations, luminosité, *etc.*) l'intérieur de l'habitacle (température intérieure, bruit), mais également la destination vers laquelle le véhicule ce dirige (horizon de conduite).

CONTEXTE EXTÉRIEUR AU VÉHICULE Ces données ne peuvent pas être rassemblées directement depuis les informations disponibles dans le véhicule parce qu'elle nécessite d'accumuler des données de plusieurs véhicules (*e.g.* le trafic routier).

### *Raisonnement par logique floue*

Les différents types d'informations contextuelles peuvent donc être organisés dans une ontologie (comme présentée en annexe *p.*221), mais les concepts logiques d'une ontologie ne peuvent être traités que par une logique déterministe. Il est donc nécessaire de trouver un lien entre des informations issues de capteurs pouvant prendre des valeurs dans un espace dense infini, avec des objets logiques par essence discrets. Nous avons donc décidé d'utiliser la logique floue pour calculer le degré d'activation de chacun des concepts au moyen d'une fonction d'appartenance qui transforme une valeur continue, en degrés d'activation de plusieurs variables logiques. Par exemple on peut distinguer trois types de destinations : près, moyennement près et loin.

De la même manière, on définit trois types d'importance pour les différents types de métadonnées définissant une liste de lecture (genre, ambiance, époque comme nous les avions définis pour la musique dans le chapitre précédent en y ajoutant les sujets d'informations pour les podcasts) : important (le filtre doit faire partie de la liste), secondaire (peut faire partie de la liste), négatif (le filtre ne doit pas en faire partie).

Nous permettons ensuite de définir des règles du type SI Destination EST Près ALORS Musique Pop EST Important ou encore SI Heure EST Midi ALORS Sujet Sport EST Secondaire. On peut alors agréger les valeurs des capteurs et calculer une valeur d'activation de chacune des règles, accumuler les valeurs pour chaque type de métadonnée de définition de liste de lecture, et obtenir les valeurs à appliquer au générateur de liste de lecture.

### *Persistance des préférences dans une ontologie*

L'avantage, de ce système d'expression des préférences est qu'il nous permet d'exprimer les préférences utilisateur dans une syntaxe à partir de données sémantiques des ontologies de contexte et d'infodivertissement. Il permet également de modifier les valeurs des poids des différentes règles dans la procédure d'accumulation, au fur et à mesure que l'utilisateur utilise le système. C'est un apprentissage limité qui ne permet pas de découvrir de nouvelles règles, mais qui permet un ajustement de l'ensemble des règles en fonction des spécificités de l'utilisateur.

### *Balisage du contenu*

Afin de permettre la diffusion de publicités localisées dans *Personal Radio* qui fonctionne de manière asynchrone entre la création de la liste lecture et sa consommation, nous avons décidé d'opter pour un balisage GPS des publicités. Ainsi, il est possible de créer un espace publicitaire dans une liste de lecture (par exemple une publicité toutes les 15 minutes), et pour chaque espace le raisonneur a le choix entre une publicité dite 'standard' et une publicité dite 'localisée'. Chaque publicité peut donc recevoir un paramètre de localisation ainsi qu'un rayon de telle sorte que si la voiture se trouve à l'intérieur de ce dernier, la publicité localisée est préférée à la publicité standard.

#### *Intégration au système Personal Radio*

L'intégration au système *Personal Radio* reprend les composants décrits dans les précédents chapitres en ajoutant de nouveaux composants fonctionnels:

- Dans le client : Nous réutilisons le client de *Personal Radio* en lui adjoignant un composant capable de rassembler les données contextuelles et de les transmettre au serveur terminal. Un raisonneur déterministe basé uniquement sur la position GPS du véhicule a été ajouté pour permettre de modifier l'ordonnancement de la liste de lecture, si des publicités contextuelles y sont présentes.

- Dans le serveur terminal : Le serveur terminal comprend les composants de création de listes de lecture étendues par rapport au chapitre précédent (ajout de la capacité de créer un mélange information/musique). Par ailleurs, nous avons ajouté un raisonneur contextuel capable d'appliquer l'algorithme décrit précédemment afin d'obtenir automatiquement les critères de définition de la liste de lecture, ainsi qu'un dispositif de sauvegarde du profil utilisateur. Les différentes ontologies servant à la définition du modèle de contexte de véhicule et du modèle sémantique de critères de définition de liste de lecture sont également mises à disposition dans une base de données relationnelle MySQL étendue pour servir de persistance à une ontologie OWL.

- Dans le serveur frontal : Une interface graphique a été définie pour permettre à un utilisateur ou à un expert de définir des préférences contextuelles en associant par 'glisser-déposer' des critères de listes de lecture, à différents concepts de description du contexte. Ces préférences sont également sauvegardées sous forme d'instances de classes ontologiques dans une base de données relationnelle. Par ailleurs, l'interface de définition des campagnes publicitaires a été étendue, afin de pouvoir ajouter des informations de localisation du contenu.

#### *Conclusion*

La croissance de la complexité des fonctionnalités d'infodivertissement des véhicules est un atout, car de nombreuses opportunités de personnalisation passive vont offrir aux concepteurs des autoradios de demain. Le Program Director que nous avons élaboré dans ce chapitre est une approche globale des différents problèmes que nous avons identifiés : (1) un modèle de contexte pour spécifier et définir l'information contextuelle et permettre une persistance des préférences utilisateur ; (2) une méthode de raisonnement capable de prendre en compte des valeurs issues de capteurs divers ; (3) une architecture adaptée aux besoins d'un véhicule.

## 2.6 ÉVALUATION DES CONCEPTS

Les différents concepts que nous avons développés recouvrent un grand nombre d'aspects dans les domaines du multimédia, des télécommunications et de l'informatique. Bien qu'une étude globale de toutes les méthodes de validation d'une application multimédia va bien au-delà du cadre de cette thèse, nous allons évaluer les différents prototypes présentés jusqu'ici à la lumière des différentes méthodes, spécifiques au véhicule et non spécifiques au véhicule de la littérature.

### *Discussion sur les différents moyens d'étude à notre disposition*

La littérature distingue les méthodes d'évaluations suivantes :

- Évaluation de la performance dans l'exécution de la tâche : Ces méthodes consistent à évaluer la facilité d'exécution d'une tâche spécifique en mesurant par exemple le temps global nécessaire, ou en utilisant un mécanisme d'occlusion visuelle avec diverses fréquences.
- Évaluation de la compétition entre tâches : L'infodivertissement appartient aux tâches dites tertiaires dans le domaine de la conduite, il peut donc être intéressant d'évaluer les tâches associées à son utilisation en compétition avec d'autres tâches (spécifiques à la conduite ou non). Ceci peut se faire, en simulateur par exemple en évaluant la capacité du conducteur à suivre une trajectoire de conduite idéale tout en remplissant une liste tâches prédéfinie, ou par une étude sur route dans laquelle on observe conducteur.
- Évaluation heuristique et qualitative : La recherche en interface homme-machine a mis au point des mesures standards comme les GOMS permettant de décrire une tâche en termes d'opérations élémentaires et de connaître la charge cognitive correspondante. Pettitt [2008] a tenté d'évaluer la capacité de prédire la performance par les méthodes précédentes en utilisant des KLM qui sont une simplification des GOMS. Par ailleurs, des méthodes d'évaluation qualitative par les utilisateurs permettent d'évaluer de manière systématique l'approbation des utilisateurs.

Le choix d'une méthode parmi celles que nous venons de citer se fait suivant plusieurs critères : Tout d'abord, la spécificité très variable de ces méthodes aux applications pour l'automobile peut conditionner leur validité ; Certaines cherchent à valider l'efficacité de l'interface, d'autres son impact sur la conduite du véhicule ; Certaines sont très standardisées et génériques et ont l'avantage de produire des résultats permettant de comparer entre eux différents prototypes. Nous avons donc décidé de nous tourner vers une étude directement en véhicule et sur route afin de pouvoir évaluer qualitativement la capacité de l'interface développée à proposer une solution efficace de personnalisation active à la fois sur le plan de la charge cognitive en observant l'usage du conducteur, et à travers la méthode dévaluation Attrakdiff que nous allons détailler.

### *Réalisation d'une étude sur route*

La procédure d'évaluation qui se déroulait dans une BMW série 7 comprenait deux phases :

- Une phase en stationnement : Le véhicule à l'arrêt, les participants devaient prendre connaissance de l'interface sans la moindre indication suite à quoi on leur expliquait les différentes fonctionnalités et on leur demandait de répondre à premier questionnaire qualitatif AttrakDiff.

- Une phase de conduite dans la banlieue de Munich mélangeant zone urbaine et périurbaine avec route et autoroute, pendant laquelle les participants devaient accomplir un certain nombre de tâches (création d'une liste de lecture avec plusieurs filtres, rejouer un titre de l'historique, jouer un autre titre du même album) tout en conduisant.

L'étude qui dure environ 45 minutes par participant à été conduite sur 18 personnes, mais en raison de problème sur le routeur UMTS dont nous disposions, seulement 11 ont pu effectuer l'étude sur route dans sa globalité et nous a permis de recueillir de nombreuses informations et suggestions : Le mécanisme de prévisualisation n'a pas été complètement compris par les participants qui ont pensé qu'il était possible de sélectionner les résultats affichés pour créer la liste de lecture. Nous avons également remarqué que les participants utilisaient les arrêts aux feux rouges pour jouer avec l'application et pour modifier les listes de lecture ce qui nous conforte dans notre choix d'un système visuel simplifié basé sur une architecture orientée sur la réactivité. Par ailleurs les résultats de l'étude AttrakDiff dénotent une augmentation des valeurs qualité pragmatique et hédonique après la conduite ce qui évoque une application particulièrement adaptée à un usage automobile.

### *Évaluation de l'architecture*

Nous avons également évalué la capacité de l'architecture à délivrer une liste de lecture en simulant différents types de charges sur le serveur par des clients se connectant sur le serveur suivant une loi de Poisson d'espérance 10 ou 60 secondes. Nous observons que la fréquence d'arrivée des clients à une influence plus importante sur le calcul de listes de lectures que leur nombre et que la prévisualisation n'est pas dégradée par la montée en charge. En effet, le calcul de liste total nécessite un accès à des fournisseurs de métadonnées qui ont des temps de latence variables alors que le calcul de la prévisualisation se fait uniquement sur les données préchargées. Toutefois le préchargement des données a un désavantage, il réduit le nombre de morceaux de musique que l'utilisateur peut découvrir lors de la phase d'agrégation. Nous avons donc décidé de mesurer la qualité des données préchargées suivant leur 'hotttness' et leur familiarité, la première étant une sorte de dérivé temporel de la seconde (*i.e.* plus un artiste devient populaire). En représentant la distribution de ces valeurs pour notre échantillon, nous avons pu constater l'équilibre entre les 'hits' et la 'longue traine'. Enfin nous l'évaluation systématique de notre système sensible au contexte posait le problème de données de références. La littérature utilise principalement des études de cas ou des heuristiques pour évaluer les architectures. Nous avons listé les observations suivantes :

1. Traitement de tout type de données de contexte : Notre contrôleur basé sur la logique floue permet d'utiliser un grand nombre de variables contextuelles issues de capteurs. Cependant, certains concepts comme le type de destination (travail, domicile, vacances, *etc.*) ne peuvent être qu'approchés par des prédicats logiques sur plusieurs variables.

2. Recommandation de tout type de données multimédia : Le plateforme logicielle que nous avons présentée n'est pas spécifique à un certain type de données multimédia et se fonde uniquement sur les métadonnées.

3. Dépendance entre la méthode d'adaptation au contexte et la fonctionnalité adaptée : Il n'y pas d'indépendance entre la méthode et la fonctionnalité. La géolocalisation et la création contextuelle de listes de lecture utilisent des méthodes d'adaptation différentes.

4. Dimensionnement du logiciel et de l'architecture : En fonction du type d'adaptation la quantité d'information contextuelle et l'importance de la charge de calcul ne sont pas les même. C'est pourquoi nous avons décidé de répartir entre le véhicule et le serveur terminal les charges de calcul et d'agrégation de données contextuelles.

5. Rôle de définition des préférences contextuelles : Nous avons envisagé plusieurs possibilités pour la définition des préférences. Ou bien un utilisateur peut utiliser l'interface web du serveur frontal pour créer des préférences simples par glisser-déposer, ou bien il est possible de modifier l'ontologie des préférences en utilisant un éditeur d'ontologie. Cette dernière solution serait toutefois réservée à un usage par un expert qui définirait des préférences par défaut.

# 3

# CONCLUSION ET RECHERCHES FUTURES : UBIQUITÉ ET TRANSPARENCE

Cette thèse traite de l'intégration de services de média à la demande dans un véhicule. Les services internet doivent être adaptés avant d'être intégrés dans les logiciels embarqués d'une voiture. En effet, ces derniers font partie d'un système complexe aux exigences variées. Nous nous sommes focalisés sur les capacités d'un véhicule en termes de connectivité, de ressources de calcul et d'ergonomie.

## 3.1 CONTRIBUTIONS DE LA THÈSE

### *Contribution en matière d'architecture de distribution de contenu*

Nous avons présenté notre solution *Personal Radio* basée sur un client riche qui supporte différents types de codecs vidéo et audio et qui est intégré aux dispositifs actuellement utilisés pour les interfaces homme-machine des véhicules. Nous avons présenté une architecture en plusieurs couches intégrant :

- Un serveur terminal qui a un rôle de couche d'abstraction pour minimiser la complexité du client
- Un serveur frontal qui a un rôle d'administration

### *Contribution à la conception d'applications réactives dans le domaine automobile*

Nous avons conçu un prototype de navigateur pour catalogue musical autour des modalités de l'interface d'un véhicule. Avec ce navigateur, le conducteur peut avec un faible nombre d'interactions créer une liste de lecture hautement personnalisée et commencer instantanément à l'écouter.

### *Contribution à la conception d'architectures sensibles au contexte*

La personnalisation passive est également une solution à laquelle nous avons proposé avec le *Vehicle Program Director* une solution. Nous avons développé une méthode de raisonnement qui permet d'utiliser des données issues de différents types de capteurs et de les relier à des préférences utilisateurs exprimées sur un niveau logique. Le *Vehicle Program Director* a été intégré à l'architecture de *Personal Radio* afin d'illustrer le potentiel des architectures de type 'proxy'.

## 3.2 TRAVAUX FUTURS

Nous pensons que les points suivants sont de potentielles futures directions de recherche dans les domaines suivants :

### *Transparence d'utilisation*

La transparence est un problème clé qui ne peut que se complexifier avec la croissance du nombre d'applications embarquées. Nous distinguerons deux aspects :

1. Les interfaces contextuelles : En fonction du contexte (*i.e.* le véhicule à l'arrêt ou en train de rouler ou de la connectivité internet, certaines fonctionnalités pourraient ne pas être disponibles ou se retrouver dégradées ou même doivent être rendues indisponibles. Par exemple la création d'une liste de lecture personnalisée lorsqu'aucune connexion n'est disponible devrait pouvoir toujours fonctionner, mais avec du contenu local. C'est un défi de continuer d'informer l'utilisateur de manière dynamique et transparente de ces changements, sans être intrusif et sans provoquer de frustration.
2. La vie privée de l'utilisateur : Dans notre architecture sensible au contexte, les préférences de l'utilisateur sont échangées en ligne avec le serveur terminal. Bien entendu, d'autres interactions avec le domicile de l'utilisateur ou d'autres dispositifs mobiles peuvent être envisagées comme un scénario de multidistribution du contenu entre différents terminaux. Toutefois, la dissémination des préférences de l'utilisateur et de ses données contextuelles entre plusieurs entités doit rester contrôlable.

### *Sensibilité au contexte*

De l'automatisation des tâche grâce à l'agrégation de données contextuelles naissent de nouveaux défis techniques.

1. Le contrôle de l'utilisateur sur les tâches : Automatiser les tâches ne doit pas priver l'utilisateur de sa liberté de piloter l'application. Réduire la complexité des interfaces contextuelles ne doit pas réduire le contrôle.
2. La sémantique des capteurs : Les valeurs de capteurs que nous utilisons et que nous 'sémantisons' dans notre module de raisonnement sont pour l'instant arbitraires et ne sont pas encore standardisées ce qui rendrait dans le futur la collaboration de différentes entités capables de raisonner difficile.

### *Ouverture*

Utiliser l'infodivertissement afin de réduire la fatigue, ou tout autre type de comportement dangereux dans un véhicule est une fonctionnalité séduisante. Toutefois il ne faut pas oublier que l'infodivertissement est avant tout une manière d'apprécier la conduite et il y a fort à parier que les fabricants automobiles continueront d'intégrer des applications toujours plus riches et complexes. D'une manière générale, la réduction de la charge cognitive continuera d'être un axe majeur de recherche dans le développement des interfaces utilisateurs et la sensibilité au contexte offre un champ large de solutions. Cependant, ce qui rend l'infodivertissement si attrayant c'est bien les possibilités d'interaction qu'il offre. C'est pourquoi l'effort de transparence pour l'utilisateur doit chercher à lui donner autant que possible le juste contrôle sur ce qu'il écoute et ce qu'il regarde. C'est une condition importante pour aider les utilisateurs à oublier leur vieux lecteur CD et à passer aux médias connectés.

# Part II

# DISSERTATION

# 1

# INTRODUCTION

> *" Before the advent of cable, American television viewers had the three networks from which to choose. In large cities, there were up to a half dozen additional local stations. When cable first came on the scene, its primary function was to provide better reception. Then new stations appeared, slowly at first, but more rapidly as time went on. Now, there are 200 or more (my cable provider offers 270), not counting the on-demand movies we can obtain with just a phone call. If 200 options aren't enough, there are special subscription services that allow you to watch any football game being played by a major college anywhere in the country. And who knows what the cutting-edge technology will bring us tomorrow."*
> — Barry Schwartz, *The paradox of choice, why more is less, 2004*

A few days after winning the American presidential election in 1992, Bill Clinton and its administration started to think about the means to support the spread of connected media as an industry. At the time, internet did not exist as we know it today. It was a network of computers linking universities and research institutions. In the mass market, the television was still the dominant media and after the boom of the cabled networks in the 70's and the 80's, the satellite technology was spreading fast to reach always more viewers. People would buy CDs of their favourites singers, and VHS of the last films they heard about. Finding, buying, consuming, and distributing content was tedious and sometimes discouraging.

The initiative taken by the young vice-president Al Gore, was about to change the face of the industry and has still consequences twenty years later in the consumption of media assets. Digital commerce has introduced a new sort of media consumption: the media on-demand. Nowadays consumers can download millions of music tracks, billions of videos from the internet in just some mouse clicks. In the meantime, the so called 'information superhighways' have spread to telephones and created mobile networks, which have themselves been integrated to private mobility devices: vehicles.

## Contents

## 1.1 PROBLEM STATEMENT: MULTIMEDIA APPLICATIONS NEED TO BE ADAPTED TO THE CONSTRAINTS OF VEHICLE PLATFORMS AND VEHICLE MOBILITY

The objective of this thesis is to make a contribution in the research field of multimedia applied to automotive scenarios. The notion of multimedia is commonly replaced by the noun infotainment in the car domain and most of the consumption scenarios of this thesis concern the consumption of audio entertainment assets and especially music since it is the most consumed media in vehicles. However, making internet media available in the vehicle involves adaptations at different stages:

### 1.1.1 *Adaptation in the architecture*

Vehicles are complex systems which involve subsystems with high security requirements and have therefore very long development life cycles; conversely, the internet ecosystem is characterised by a high volatility of the services. The architecture of a vehicle media consumption framework has to take this requirement into account in order to provide robust solutions which are flexible enough to be quickly adapted.

### 1.1.2 *Adaptation in the design*

Consuming media in a vehicle is definitely different from consuming media on a TV-set, on a computer desktop or with a MP3 player. The ergonomic of the application involves not only the modalities which are to be used in order to interact with it, but also the architectures of the software design in terms of efficiency and transparency to the user.

### 1.1.3 *Adaptation of the media selection*

With the increasing amount of media which is made available to the user, their selection can introduce a cognitive workload which can be reduced by assisting or forecasting the choice of the user, depending of the context. Indeed, adapting the design of application interface is not always sufficient and the selection of media has to be adapted to the driving situation.

### 1.1.4 *New scenarios, new concepts, new solutions*

As a matter of fact the automotive integration of existing media services pertains to much more than a mere adaptation. Those are completely new architectures and services which have to be created.

#### 1.1.4.1 *Efficient aggregation and caching of metadata*

The diversity of information about media is a key to the selection issue. Metainformation coming from multiple sources must be consistently aggregated and delivered to the mobile media client which has limited ressources.

#### 1.1.4.2 *Distributed context user reasoning*

Supporting the decisions of the user, or forecasting his intentions need a computational but also architecture model for the deployment of reasoning capacities among the different entities (media provider, media receiver), and a persistency model for the dissemination of those preferences.

## 1.2 RESEARCH METHOD: CONCEPTION PROTOTYPING

Not only does this thesis give new solutions to the problement statement in terms of industrial innovations, the purpose of a doctoral thesis is to bring scientific contributions.

### 1.2.1 *Prototyping*

All the concepts presented in this thesis have been developed both after a thorough study of the state of the art (incremental research), and based on discussions with specialists of the different research fields involved in the problem statement (exploratory research). Most of this thesis was conducted at the architecture team of BMW Research and Technology and research methods had to be adapted to the working habits of an industrial team used to making 'demonstrators'. They consist in implementing so called 'proof of concepts' to demonstrate the technical feasibility of the architecture.

### 1.2.2 *Experimental evaluation*

Some of the former solutions have been tested with experimental setups involving a true prototyping phase. The evaluation of the user-interface application design was conducted together with another BMW department specialised in human interfaces.

## 1.3 OUTLINE OF THE THESIS

This thesis is divided into three parts that we summarize here: This two first chapters introduce the different fields involved in this thesis: Consumption scenarios of online media, vehicle embedded systems and recommendation of on-demand content.

- *Chapter 2 – Mobile Media consumption and Vehicles: Physical and Virtual Paradigms*. This chapter defines the problem of vehicle media consumption as a user-centric issue. We analyse the different steps of an on-demand media selection story line and identify the key aspects of user personalisation. We also present the new types of services that have emanated with the online media content. We introduce the technical functionalities that we have reused in the solutions proposed further in the thesis.

- *Chapter 3 – The digital media revolution, metadata and recommendation*. This chapter explains how the metadata which have appeared together with the digitisation of media assets can be put to good use to support the user in his media usage. Generalities on metadata are followed by an analysis of the metadata extraction and production workflows. Finally, we present the media recommendation and discuss the feasibility of its integration in a vehicle scenario.

The chapters 4, 5 and 6 answer the core questions of the problem statement: How to deliver media assets in a vehicle? How to enable the personalisation of content? and split the answer of individualisation into two sub-problems: user active and user passive personalisation.

- *Chapter 4 – On-Demand Media Delivery, The Personal Radio Architecture*. This chapter explains the choice of a multi-layered web architectures for the delivery of on-demand media assets. We first compare the main types of media delivery: broadcasting, central service and routing, peer-to-peer. This chapter then presents the implementation of the personal radio architecture which combines a rich client with a service abstraction. The flexibility of the multi-layered architecture is illustrative with a prototyped study of personalised product information through the connection with a CRM system.

- *Chapter 5 – Interactive user personalisation, Browsing music recommendations in a vehicle.* This chapter is an analysis of the ergonomics of a vehicle infotainment application through a prototyped study: a music browser. We analyse the different modalities which can be combined for user input and output and the different types of architecture alternatives to provide recommendation to the user and support his active choice. The chapter then presents the implementation of the browser through its different functional and technical components.

- *Chapter 6 – The context-driven user-passive personalisation.* This chapter presents an alternative to the former solution. Instead of asking the user to interact with the application, we can use context information to infer automatically his wishes. We first analyse the problem of context sensitive systems and the different types of architectures regarding the context information access, reasoning and mapping to multimedia application. The chapter depicts a framework based on fuzzy logic that we have implemented and adapted to the personal radio architecture and which gives solutions to the following issues: context user preference persistency, context-based remote media selection and geo targeting media display.

This two last chapters evaluate the different solutions proposed in the previous chapters and draws the future research outlook.

- *Chapter 7 – Evaluation of the Concepts.* After reviewing the very rich literature on evaluation methods for the vehicle infotainment solutions. We present a qualitative study of the music browser based of the chapter 5 on the Attrakdiff method. We also present measurements made on the architecture. Afterwards, we heuristically review the context-driven solution presented in chapter 6.

- *Chapter 8 – Conclusion and Future Works.* This chapters summarises the results of the thesis and presents the future research directions.

# 2

# MOBILE MEDIA CONSUMPTION AND VEHICLES: PHYSICAL AND VIRTUAL PARADIGMS

> *" We now know media use to be generally a very untidy, inefficient, and chancy matter, with multiple meanings."*
> — Denis McQuail, *Audience Analysis, 1997*

> *" Any customer can have a car painted any colour that he wants so long as it is black"*
> — Ford & Crowther, *My life and work, 1922*

The consumption of media is nowadays guiding the design of most of the products sold on the consumer electronic market and the vehicle headunits are no exception. The offer has itself dramatically changed with the digitisation of media. This chapter depicts the current trends in media consumption and delineates the relationship with our problem statement: The personalisation of the infotainment vehicle experience. This chapters first deducts from a vehicle scenario the functional requirements of a peronalisation solution. Afterwards, we analyse the media consumption process and systematically defines the different steps of the multimedia experience. Then, we analyse the different types of media services that have arisen in the previous decade and what is the place of the vehicle in the personalisation chain.

## Contents

## 2.1 ANALYSIS OF THE VEHICLE INFOTAINMENT SCENARIO

The integration as such of services on-demand, does not make sense for a vehicle because of the usage specificities. We first need to imagine a usage scenario that depicts the general need of vehicle users regarding entertainment and information. The solutions that we will develop in this thesis are linked to the following scenario that helps us to identify the role of the vehicle in the whole infotainment ecosystem.

*General Scenario*

Alice is married and has two children, she is living in a big city and needs to drive between 20 to 30 minutes to commute to her work at a newspaper in the morning and roughly the same to go home in the evening. She leaves her home between 7:30 and 8:30 and goes home around 18:30, depending on the amount of work she has and if she has to drop or pick up the kids at school. Alice likes listening to music when driving but also need to be up to date with politic and economic news for her job. Therefore, she listens to information when driving in the morning but prefers to relax with music she likes when coming back home. Since she lives in a city, her driving is often interrupted by firelights, and sometimes traffic jam. Alice has very changing music tastes depending on her mood, the season and she likes to show her colleagues her new findings when they go out for a lunch or for a drink after work. Like most people driving a private car, Alice belongs to the working population and has therefore a purchasing power of interest to advertisers. Alice does not mind receiving ads when driving as long as she is not disturbed and that the ads are not overwhelming.

### 2.1.1 *Access to content on-demand*

In this scenario, it is quite clear that Alice has a changing schedule, therefore depending on the moment when she starts driving she might miss the political review or information flash of a radio broadcaster. Moreover, Alice has specific music tastes and wants to personalise the music playlist. In brief, Alice wants to have control on the program that she is listening to. We will call this functionality the content on-demand, which is a functional requirement for the development of the solutions that are described in this thesis. Alice may want to store online her music preferences and update her profile with her listening habits so as to carry on interacting with the social networks while driving.

### 2.1.2 *Reactive user interface*

When driving Alice has several opportunities to specify her preferences to the system. She finds traffic lights on her way or may be stopped by a traffic jam. Besides, when she parks in front of the school to pick up her kids, she might want to search for music or videos for her children. An interactive user interface using both visual and haptic modalities can be used in order to browse through the online choice of content but this interface has to be reactive enough not to disturb the driver.

### 2.1.3 *The use of the context to help the user*

Even if the schedule of Alice is changing, it is possible to find a general pattern in her media consumption and to correlate this with her driving habits. Alice does not want the same program whether she is driving in the morning, or in the evening. The multimedia application can adapt the choice of the content and create automatically a personalised program depending on the context parameters of the drive.

## 2.2 MOBILITY AND INFOTAINMENT

The evolution of transportations has always shaped societies, and the spread of private transportation has recently accelerated the mutation of urban regions. In 2004, a study lead by the federal telework Exchange [2005] noticed that in the USA an "average Federal employee who commutes five days a week to his work spends 245 hours of his life per year going to his work"; which means that he drives around 5 hours per week. Ausubel *et al.* [2005] who made a meta study of different user travelling survey came also to the conclusion that one hour per day on the daily commute is a fair approximation, but also notices strong variations between the countries (USA, Japan, Germany). Infotainment and listening to music is a mixed blessing. Dibben and Williamson [2007] point out in a metastudy that there was a correlation between accident occurence an the use of an infotainment system. In the same study they notice that "listening to music is commonly mentioned as a counter-measure to driver fatigue". Since commutation time increase, the need of infotainment to entertain and inform the user is also augmenting. Therefore, the development of media application is necessary but also has to bear in mind security concerns.

People mobility has changed, but the media landscape experiences a mutation too. Therefore, we need to understand what are the new media consumption process and how they are coupled with new types of media distribution.

## 2.3 UNDERSTANDING THE MEDIA AUDIENCE: THE MEDIA CONSUMPTION PROCESS

The media consumption has for long been a research interest for behaviorists in the sociological and economic fields trying to ascertain what

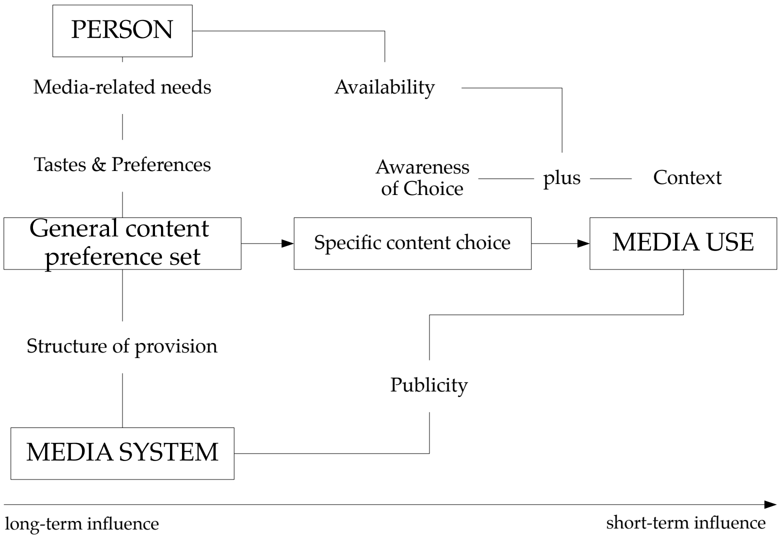
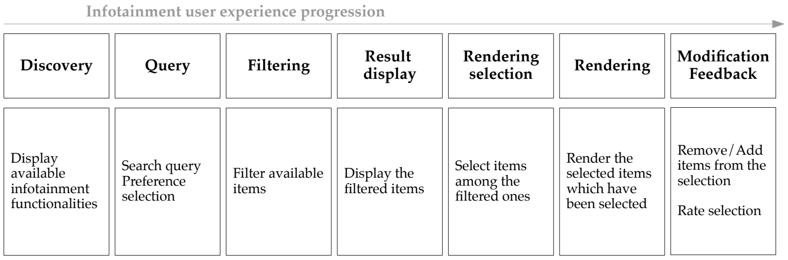


Figure 3: An Integrated Model of the Process of Media Choice[McQuail, 1997]



| Discovery | Query | Filtering | Result display | Rendering selection | Rendering | Modification Feedback |
|---|---|---|---|---|---|---|
| Display available infotainment functionalities | Search query<br>Preference selection | Filter available items | Display the filtered items | Select items among the filtered ones | Render the selected items which have been selected | Remove/Add items from the selection<br>Rate selection |

Figure 4: The different steps of the consumption of on-demand media

are the influencing factors in the user choice. Seufert and Ehrenberg [2007] summarized the problem to the two following questions: (1) “are the decisions on the use or non-use of certain media or content categories made in an active conscious and rational matter” or (2) “ are they passive, unconscious and habitual or depending on mood”. This problem is also know as a “lean back / lean forward” dilemna [Zaletelj *et al.*, 2007].

The model of McQuail [1997] (as depicted in figure 3) introduces the notions of long-term and short-term factors in the choice of media assets. Three concepts turn out to have a strong influence in the media consumption: (1) Awareness of choice, (2) Available Options and (3) User Context. The aspect of awareness of choice and of active customizing will be tackled more into details in the chapter 5 of this thesis, and the context aspect in chapter 6.

### 2.3.1 *The different steps of the on-demand media consumption*

The model of McQuail depicts the consumption process from a rational user point of view. We can apply it to the consumption of on-demand media. We describe the consumption of on-demand media as following (see figure 4):

DISCOVERY This is the first step of the multimedia consumption experience for every type of device, when the system invites the user to formulate his query. Vehicle infotainment systems combine haptic modalities (buttons, commands, touch screen) with a visual modality (a screen).

QUERY FORMULATION In this step the user formulates his query. Depending on the user interface it can be a simple action to start to "play" or the selection of tracks and filters that have to be matched. (Play similar music to an artist, play podcast from a specific date, etc.).

FILTERING Devices generally do not display this step to the user but plays an important part in the personalisation process. The system has to select content which matches the query of the user. If the system has a previous knowledge of the preferences of the user, it can use it to increase its chances of filtering a relevant output.

RESULT DISPLAY At this point the result of the filtering is displayed to the user. It is not compulsory to display the whole result, we believe that the display of a preview is a good trade-off between HMI principles and resource efficiency, since a preview of the result can help the user to decide whether he wants to consume the whole output or go back to the query step. Some systems start the rendering together with the display, and when different solutions are proposed it is possible to propose a selection step.

RESULT SELECTION This is an optional step, since some systems directly skip to the rendering after the display of the results.

RENDERING In this step the user consumes the result of the filtering. Audio and Video output is used to playback the files.

RESULT MODIFICATION This step, which is also optional, is very important when addressing the field of mobile multimedia applications for which the devices provide limited interactions. Indeed, we have already listed up to four steps in the multimedia experience, between the discovery and the rendering of the results. If the user changes his mind, it is very important that he can modify the result without having to restart the query formulation from the beginning.

### 2.3.2 *Short/Long Term, Active/Passive personalisation*

In the aforementioned steps the decision of the user can be influenced by more-or-less short-term and long term factors. The tastes, preferences, and interests are long-term user preferences, but on the other hand the mood and the context can highly influence the type of choice that the user is making. The chapter 6 discusses these aspects and presents a solution to the management of context information to support the choice of the user. McQuail also noticed that the 'availability' (we prefer the term of 'discovery' when describing the consumption process) shapes the audience choice. The more options are proposed, the more actively a user will participate to the formulation of a preference. The notion activity in the development of vehicle media application has a special importance as we will see in chapter 5, since the infotainment

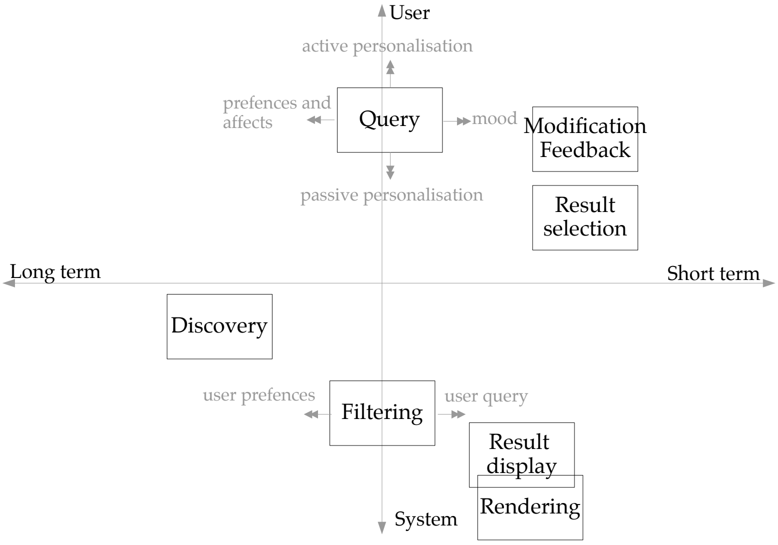


Figure 5: An example of the distribution of our consumption steps model using the scale of McQuail

related tasks yeld a cognitive load competition with other driving-related tasks. In the figure 5 we have intentionally placed the filtering step on the infotainment system side, since we think that this step has such a cognitive load, that it should not be a user task at all in a vehicle infotainment system. Of course, there are other media systems which leave the filtering to the user.

## 2.4 THE ACTORS OF THE MOBILE MEDIA INDUSTRY AND THE DIGITISATION OF MEDIA ASSETS

In the domain of vehicle infotainment, the vehicle appears to be the end device of a complex value chain.

### 2.4.1 *Digitisation of the offer*

The amount of digital media has kept on growing in the past decade (see evolution of the digital digital music market figure 6). The Youtube service launched in 2004 which gives the possibility to share videos between users has gained a tremendous popularity, indeed “every minute, 24 hours of video is uploaded to YouTube”[1]. The illegal copy and download of digital media especially music and films, has also accompanied the arrival of digitisation. In 2004 there were 900 millions of music assets infringing copyrights shared by 200 millions of users [IFPI, 2009].

### 2.4.2 *Media infrastructure: Growth of mobile on-demand infrastructure and cloud services*

In order to provide mobile access to the digital offer, mobile internet service providers (internet service provider (ISP)s) have developed two

---

[1] http://www.youtube.com/t/fact_sheet

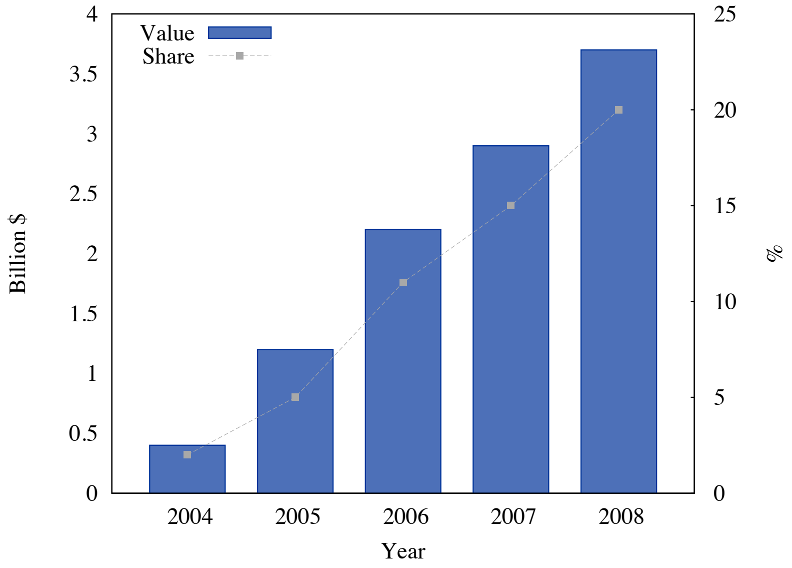


Figure 6: Evolution of the music digital market [IFPI, 2009]

aspects of their business: On the one hand they have densified their wireless network with new mobile technologies, but moreover they have developed so called content delivery networks (see 7) , to speed up the access to on-demand content (the section 4.2.2 gives a more technical definition of content delivery network (CDN)s).

A third infrastructure aspect is the sometimes very hackneyed and blur notion of cloud. Indeed, it is a fact, as we will see in this thesis that developing on-demand services creates a growing need for computational resources. The idea to share the design of the logic of an online application and the scaling of its processing runtime (computational, storage and network ressources) has contributed to apparition of 'storage providers' or 'computing resource providers' *etc.* with solutions which scales automatically.

### 2.4.3 *A taxonomy of mobile media services on-demand*

The diversity of online media services does not only reside in their content but also on the type of service they deliver. For instance, in 2008 their were already more than 500 different music services available on the internet and this number is continuously growing [IFPI, 2008]. The most exhaustive study on the topic has been realised by Katsma and Spil [2010], who made a taxonomy of music services. We can extend this taxonomy of services to all type of audio and video content services provided on the web. For this Katsma and Spil, took four main criteria in consideration:" the business model, the service functionality, the supplier identification and the technology of distribution". Their study identifies four types of media services that we can extend with a fifth one (the media crawlers).

EXTENDED MEDIA BROADCASTERS They are the traditional branch of media delivery: TV, Radio now broadcast online their content using IP networks. The programs are based on an editorial schedule and are not personalised but the user can download and replay content

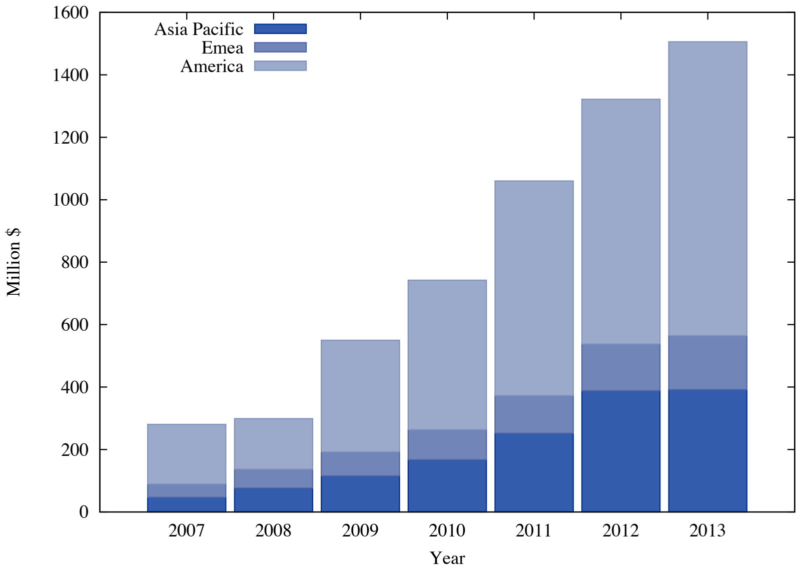


Figure 7: Prospective evolution of the CDN market, Frost & Sullivan [Global, 2009]

with podcast and catch-up services. They base their business model on advertising and sometimes subscription and they rely on a streaming/-download infrastructure. Commercial examples are: BBC, Deutsche Welle, Radio France, *etc.*.

PERSONALISED INTERNET RADIO These radios permits a user-specific content provision and use a 'recommendation mechanism' to create a radio stream personalised to the user. The user needs to specify his preferences, and a program is automatically generated. Commercial examples are: Pandora[2], Musicovery[3], *etc.*.

EXCHANGE COMMUNITY Communities are services which provide "sharing music combined with a socializing character to virtually meet". There are also video sharing communities. User actively share comments and impressions about the content and those communities provide recommendation services to discover new content. They mostly rely on websites and provide APIs to access their services (like the scrobbling API of Last.FM). There business model is based on advertising and also commissioning with the following category. Commercial examples are: Dailymotion[4], Last.FM[5], MySpace[6], Youtube[7], *etc.*

ONLINE RETAILERS The online retailers provide content on-demand (music, films, series, *etc.*) on different business models : subscription business models, pay per view or pay per download. They rely on web sites (like Amazon) or on specific software (like iTunes[8] or Spotify[9]) or

---

[2] www.pandora.com
[3] www.musicovery.com
[4] www.dailymotion.com
[5] www.last.fm
[6] www.myspace.com
[7] www.youtube.com
[8] www.itunes.com
[9] www.spotify.com

are integrated in specific home entertainment devices (film rental at Netflix[10].

MEDIA CRAWLERS The media crawlers do not emanate in the study of Katsma and Spil [2010] but they are a good exemple of the apparition of new 'cloud' actors who extend traditional search engines. Crawlers analyse the content which is available on the internet and extract knowledge from it. In the music domain, the Echonest[11] for instance can provide information about the popularity of an artist. They also provide services like content identification and additional information.

## 2.5 THE ROLE OF THE VEHICLE IN THE INFOTAINMENT VALUE CHAIN

As we can see, not only the digitisation of media did change the value chain [Rayport and Sviokla, 1996] in the distribution of entertainment content, but it also redistributes the roles of the different entities in the virtual value chain (electronic assets) and physical value chain (the vehicle). From this scenario and from the taxonomy of services that we have presented we can sketch a role model to identify what is the place of the vehicle in the value chain.

### 2.5.1 *The content provider and metadata provider*

The content providers are the entities of the media industries that assure the distribution of media over networks. Their role is to produce this content and to distribute it with a business model. The production of content goes far beyond the scope of this thesis.

### 2.5.2 *The content aggregation/selection agent*

This role is quite new, because at the time when media where directly broadcasted, there was no need for aggregation and selection. Now that the content is reaching an unprecedented diversity because of the digitisation of the offer, consumers need agents capable of aggregating it and selecting it. As we have seen in the scenario and in the technical requirements of the former chapter: this aggregation and selection is specific to the usage (in our case a vehicle usage). Therefore, we will develop in the chapters 4,5 and 6 vehicle specific media aggregation and selection methods.

### 2.5.3 *The content access and consumption*

The content access and consumption role is shared between the network providers which develop and maintain infrastructures for the deliver of assets, and the end-device producer (in our thesis, the vehicle manufacturer). The consumption of these assets takes place in the vehicle.

---

[10] www.netflix.com
[11] www.echonest.com

## 2.6 CONNECTED VEHICLES AND AUTOMOTIVE INFOTAINMENT

There is a need for media consumption in the vehicles and this need is evolving with the increasing time that people spend in individual transportation. Moreover, the development of a digital media audience has changed the role of the vehicle in the media value chain. Vehicles are no longer reduced to physical means of transportation, they are now service integrators. The past decades have seen a tremendous evolution in the embedded electronic of vehicles. This transformation has shaped new software conception methods. We will now describe the state of the art of the vehicle electronics and entertainment platforms and their specificities in terms of ressources and introduce the notion of multimedia embedded software.

The vehicles sold on the market at the end of the nighties had already as much electronic devices as the airbus aircrafts conceived in the sixties[01net, 2006] and the tendency to integrate more devices keeps on growing. In the following, we will present the technical capabilities of vehicle for infotainment applications and present the frameworks that we use to develop the applications that we present in the following chapters.

## 2.7 THE PLACE OF INFOTAINMENT IN THE VEHICLE ELECTRONIC SYSTEM

The word infotainment is composed of information and entertainment which covers very diverse functionalities:

- Media consumption: radio, CD, Music player
- Communication: telefon, email
- Connectivity: adressbook synchronization, mp3 player
- Online services: weather forecast, stock exchange information, hotel reservation

The importance of software development has been continuously growing in the last decades, together with the increasing inner connectivity (embedded systems) and outer connectivity (communications and telematics) of vehicles.The infotainment itselfcovers many two parts (multimedia, and human machine interface (HMI)) of the whole vehicle so called 'electric/electronic' software domains which are composed of: chassis, transmission powertrain, car body indoor, multimedia and human machine interactions. Nowadays, a premium vehicle has more than 700 software-based functions [Salzmann, 2009] and this number will not stop increasing. The multimedia domain is characterised in comparison with the other domains by a very high complexity of the software programs (between 100 and 300 MB of code) in comparison with other domains (*e.g.* there are only 2 MB of code for the powertrain domain) and a large number information types to control the functionalities (up to 600). Yet, the multimedia domain does not have the real-time constraints of automotive security systems.

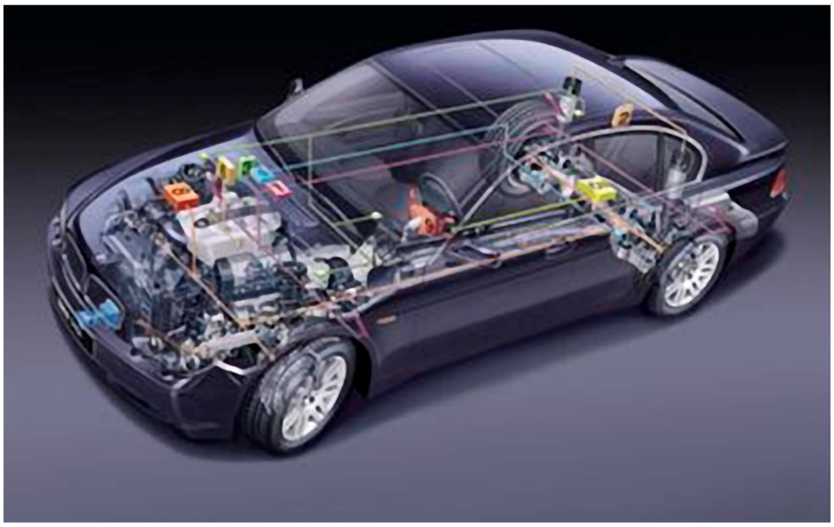


Figure 8: Vehicle control units, actuators and sensors are connected over a complex network (image BMW)

## 2.8 VEHICLE CONNECTIVITY AND VEHICLE BUSSES

With the advent of internet services, the connectivity of the vehicle plays a crucial part in the design of infotainment applications. The vehicle connectivity (see figure 8) is complex and aims at two different goals: (1) linking control units of the vehicle (*e.g.* in the transmisssion domain actuators and sensors, in the HMI domain connect the head unit to the display and the interface knobs, *etc.* and (2) connect the vehicle to the external world (receive radio and television programs, connect to the internet,*etc.*).

### 2.8.1 *Vehicle busses*

Two vehicle specific busses dominate the vehicle architecture of infotainment systems: the controller area network (CAN) bus and the media oriented systems transport (MOST) bus.

CAN The first one has been originally specified by Bosch and is today a *de facto* standard in the whole vehicle industry. CAN is not only used for the infotainment, it is also used for other domains of the vehicle. The specifications of CAN cover all the layers of the open systems interconnection (OSI) model from the physical layer to the transport layer. CAN has a bus topology which means that all the control units share the same communication line. It uses a CSMA/CA technique to avoid collisions. CAN is capable of transmitting synchron information but cannot fullfill all the requirements of a realtime bus.

CAN is very robust, and is so established that it has achieved a very good quality/price ratio. In the infotainment domain, CAN can be used when the control elements (like the buttons, or knob) have to communicate with the head-unit. However, the CAN standard support only a bitrate of 1MBit/s over 40m which is for infotainment not sufficient if it has to transport wireless internet data which use much faster bitrates (see section 2.8.3).

MOST In order to support multimedia applications which have a high demand in terms of datarate, the MOST standard has been developed by the german facturers for the middle and premium segment. MOST has been designed to transmit audio and video signals over a synchron network. MOST is a made of a ring of control units, which are linked with an optic fiber. Therefore, developing control units having interfaces with MOST is much more expensive than for CAN. MOST specifications goes up to the application layer of the OSI model. There is a catalogue of more than 600 messages to control the CD player, the screen, *etc.* but MOST can also transport packets from the IP stack. Ethernet, is also today gaining a growing interest in the vehicle industry and should in the near future provide a economic alternative to MOST. Since the control units have to communicate between them, the different busses are linked to a so-called 'gateways'.

### 2.8.2 *Vehicle radio and television*

Even before the vehicle got connected to the internet it was still possible to consume medias with vehicle infotainment systems. The first autoradio was embedded in a 'tuned' T-Ford model in 1922[GFU, 2010]. Industrialisation of the production came much later in the USA (1927) and in Europe (1932). Still today, vehicle need to be equiped with tuners for every specific analog or digital broadcast techniques but the use of software-defined radio could in the future reduce the complexity of embeded tuners.

### 2.8.3 *Vehicle mobile internet*

The arrival of mobile internet in the vehicles came quite late in comparison with the mobile internet for smartphones. Since 2007, german carmakers propose general packet radio service (GPRS) and enhanced data rates for GSM Evolution (EDGE) in their vehicles, and should in the near future integrate universal mobile telecommunications system (UMTS) and 3GPP long-term evolution (LTE).

## 2.9 EMBEDDED SOFTWARE AND HARDWARE FOR VEHICLE INFOTAINMENT

Developing infotainment applications for vehicles involves dealing with both functional and technical requirements. As we have seen in section 2.4, the amount of internet services is continuously growing creating new functional requirements. Although, the complexity of vehicle electronic is such that technical requirements tend to have longer life-cycles (*i.e.* a new head-unit may not be developed for every single new internet functionality).

### 2.9.1 *The vehicle head-unit*

Formerly composed of an auto-radio, head-units are today litterally embedded PCs. The prototype of personal radio which is presented in chapter 4 has been designed to run in a BMW vehicle equiped with a car infotainment computer (CIC)-High system and iDrive controller elements as displayed in figure 9.

| | | |
|---|---|---|
| Program size | between 100 and 300 MBytes | |
| Controll units | between 4 and 12 | |
| Bitrate | 22 MBit/s | 1MBit/s over 40m, 10kbit/s over 5000m |
| Sorts of communications | 660 | |
| Time cycles | 20ms – 5s | |
| Bus | MOST | CAN |
| Topology | Ring | Bus |
| Security requirements | relatively low | |
| Telecommunications Analog:<br>Digital: | FM and AM Radio<br>DAB and T-DMB; GPRS<br>EDGE and soon UMTS | |

Table 2: Characteristics of the vehicle infotainment platforms (extended from Salzmann [2009])

Figure 9: The infotainment controls and display in a vehicle of the premium segment (image BMW)

Figure 10: The rear seat display (image BMW)

### 2.9.2 *User Interface*

Since most of the multimedia applications are interactive applications, the user interface plays an important role in vehicle infotainment systems. It is composed of devices for the three following modalities: visual modality, haptic modality and sound modality.

THE DISPLAY Currently, the multimedia infotainment functionalities are displayed on a head-down colour display which is situated on the right of the driver (see figure 9) and which is linked to the headunit with a low-voltage differential signaling (LVDS) link. When driving some functionality of the display are greyed out like the TV programs or the internet browsing in order to avoid user distraction. Upmarkets vehicles propose also rear seat display so that rear passengers can watch films or select music from their seat (see figure 10). The resolution of the display has been growing much more than the dimensions which are limited for ergonomy reasons. Depending on the model, the resolution can is currently 800x480 pixels or 1280x480 pixels for the front head-down displayn and 480x480 pixels for the rear seat displays.

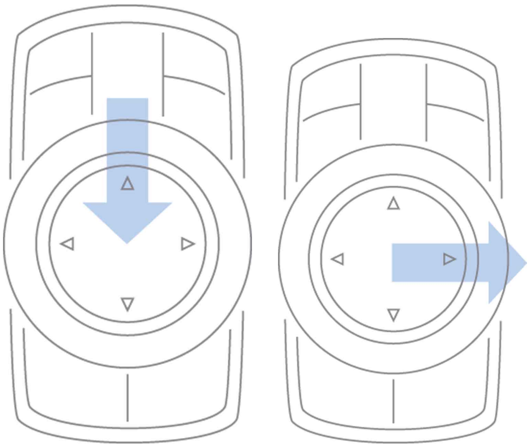
THE CONTROL ELEMENTS The control elements are essentially, knobs and buttons which are linked to the headunit with the CAN bus. On the steering-wheel there are buttons to skip CD tracks or radio stations, in the middle-console there is a knob to set the volume and buttons to select tracks (see figure 9). But an innovation of the last generation of control elements has been the development of a multifunction knob capable of different controlling movements as depicted in figure 11.

THE SOUND OUTPUT The loudspeakers are connected to the headunit over the MOST bus. The sonorisation of the vehicle has some constraints since it has to be designed for a use when driving which means, among the different problems of integration; The lossy sound compression formats which reduce the dynamic range of the audio signal can be used in vehicle since the distorsion perceived much smaller than with a high fidelity (HiFi) system Krump [2008]; The sound system has to have a sufficient sound level so that quite passages of music are not covered by the noise of the cruise (engine noise, cockpit noise, *etc.* The sound level can be adapted to the speed of the vehicle with the geschwindigkeitsabhängige Lautstärkeanpassung (GALA)[12] systems ISO [1992]. However, it is also necessary no to damage the ear of the pas-

---

[12] speed adaptative loud speaker amplification

(a) The iDrive (b) Rolling axis



(c) Pushing axis (d) Shifting axis

Figure 11: Presentation of the different controlling movements possible with the BMW iDrive®

sengers. For this, the repartition of the speakers is essential [Meroth and Tolg, 2008] (see figure 12).

### 2.9.3 *Infotainment software development for the vehicle*

The development of software for the vehicle has to deal with the following problem: (1) on the one hand, vehicle architectures have long development life-cycles (2) infotainment functionalities and especially connected functionalities have much shorter life-cycles and have to be updated often in order to follow user trends and market trends. Therefore, it is necessary to have a development framework in the multimedia domain which can:

- Abstract the different hardware specific requirements in order to achieve a separation of concerns: *e.g.* the development of infotainment applications should be independent from the development of a new version of the iDrive as long as the functionalities provided by the haptic component remain the same.
- support graphic, audio and video codecs.
- Enable to load, deploy and remove infotainment applications for a fast testing of new functionalities.

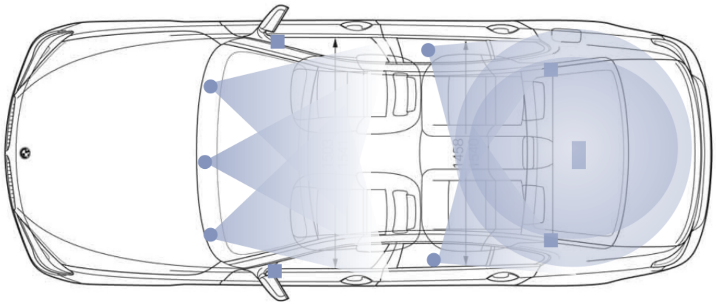


Figure 12: Repartition of the audio speakers in a vehicle of the D-segment (from [Meroth and Tolg, 2008], image BMW)

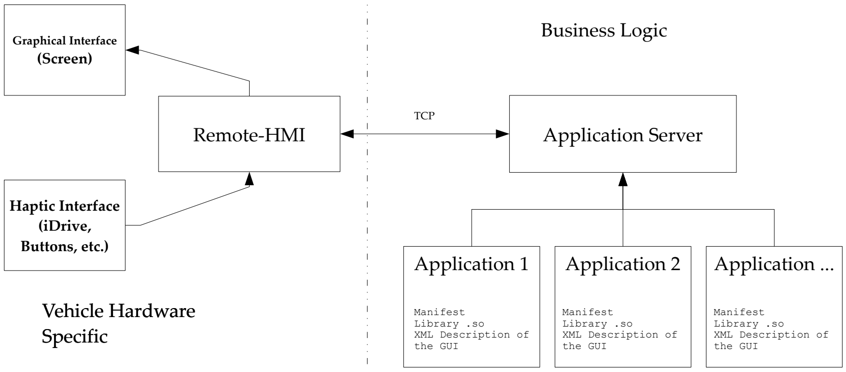


Figure 13: Separation of concerns: Remote HMI

### 2.9.3.1 *The remote HMI*

The personal radio application that we present in chapter 4 is developed with this framework. The Remote HMI which is developed by the BMW CarIT gives an interesting solution to the separation of concerns. It consists in a client/server application based on a similar model to the X Window System[13]. It consists in a client which is capable of rendering graphics and handling haptic input interfaces see figure **??**. The client and the server communicate over TCP (see figure 13).

The server can load applications which contains the business logic, those applications are C/C++ libraries which are declared by a manifest. The elements of the user interface are written in extended markup language (XML) and it is possible to develop applications based on events mechanism (*i.e.* write procedures which are called when a user interface event is triggered).

A very big advantage of the remote HMI application, is that there is a simulation of the user control elements, so that a developer can first develop on his desktop for the remote HMI and then, with the same software, test the applications in a vehicle.

---

[13] X Window system is the standard window environment for UNIX systems which "provides a client/server interface between the display hardware and the desktop environment: http://www.xfree86.org/, http://www.x.org/

#### 2.9.3.2 *A vehicle browser*

Browsers have been specifically developed for connected applications. They all support the HTTP protocol and can render vector and raw graphics, videos and sound using standard like HTML 5, canvas, or Flash. Until the advent of HTML5, browser applications were completely online applications with no possibility to store locally content. An advantage of online browser applications is that they are implemented on light clients and they can be updated just by changing the web page delivered by the server. Browser applications can be easily tested on a desktop before being deployed in the vehicle. That makes them especially suited for a fast prototyping life-cycle. At BMW a specific version of the Webkit Engine has been developed for research purpose to be interfaced with the CAN commands in order to receive user interactions. We will present in chapter 5.

## 2.10 CONCLUSION

We have established that the integration of existing information and entertainment services involves specificity to the vehicle usage. The digital media on-demand world is actually a various set of different types of services including personalised radios, online retailers and social communities. We have formally depicted the different steps of the media consumption from a user point of view and highlighted the different importance of the human factor and of the system factor can play in this process. Developing new infotainment applications for the vehicles cannot reuse traditional software development techniques for mobile phones or desktop applications without major technical adaptations. Inner connectivity and outter connectivity need to be designed in order to support online applications with high interactivity and bitrate demand. Moreover, the user interfaces for the vehicle are splitted among different devices which makes necessary to separate concerns between hardware and software development. Two solutions: the RemoteHMI and the vehicle browser are briefly introduced as solutions for the development of media applications in fast prototyping life-cycles.

# 3

# THE DIGITAL MEDIA REVOLUTION: METADATA AND RECOMMENDATION

> *" It may be objected that metalanguage also makes a sequential use of equivalent units when combining synonymic expressions into an equation sentence: A = A "Mare is the female of the horse". Poetry and metalanguage, however, are in diametrical opposition to each other: in metalanguage the sequence is to build an equation, whereas in poetry the equation is used to build a sequence."*
> — Roman Jakobson, *Closing statements: Linguistics and Poetics.*
>
> *Style in language, T.A. Sebeok, New-York, 1960*

Digital media do not only provide better render quality, easier duplication and faster transport than their analog counterparts. Their advent has seen the development of another type of information: metadata. This chapter explains why and how metadata are a significant contribution to the personalisation of on-demand media consumption. It analyses the different types of metadata which are associated to media content, the way they can be used and how they can be accessed in an embedded device. This chapters depicts the workflows for the creation, extraction, spreading and consumption of this information among the different entities of the multimedia value chain. Finally, the chapter tackles the main functional interest of metadata for our problem statement: recommendation.

## Contents

The notion of 'meta' information introduced by the linguist Jakobson, covers in the multimedia domain an increasing role. Not only does the increasing resolution of media files which can be transmitted over communication networks creates a demand for metainformation in order to classify, sort and search this content but also the boom of interlinking between those different types of content create an hyperspace of so-called 'hypermedias' which are themselves part of a 'metaverse'.

## 3.1 STANDARDS FOR THE REPRESENTATION OF METAKNOWLEDGE

Expressing information about media assets has been manyfold normalized and standardized. Metainformation extend the media content and can be processed together with it (*e.g.* display the name and the artist of a track during the playback) or separately from it (*e.g.* filter all the tracks of a catalogue according to their year of release). There are different types of metadata [NISO, 2004]: descriptive metadata are intended for searching and identifying content, administrative metadata provide information about the creation, the storage and the distribution of media assets, rights management metadata contains copyright enforcement information (*e.g.* digital rights management (DRM) information), structural metadata composition indication of media documents (*e.g.* synchronisation information between audio and video streams).

Different purposes, different needs and different applications have lead the members of the media industry to create standards for metadata [Hoffmann, 2006]. Some standards try to achieve a very precise purpose like Dublin Core (description of a media according to 15 types of metadata) or society of motion picture and television engineers (SMPTE) (labelling of video frames), european broadcasting union (EBU) Core (thesaurus for the description of TV and radio programs) some give more exhaustive frameworks like motion picture expert group (MPEG)-7 and MPEG-21 which give exhaustive description formats for all sorts of medias.

### 3.1.1 *Textual data*

Textual data is the most atomistic type of information. It can describe different types of knowledge in both human readable and computer processable forms. Obviously textual data can be text information such as the name of a media (*e.g.* title of a music track, its author or its interpret, the topic of an information, *etc.*) but textual information can also describe values (*e.g.* the date when an information podcast has been recorded, the GPS coordinates of a picture or the length of video track, *etc.*). For instance using the HTTP protocol (see chapter 4) a server can inform a client that the file it is willing to download is an audio file in an MPEG container using the `audio/mpeg` media type. The internet media types (originally called multipurpose internet mail extensions (MIME) types [Freed and Borenstein, 1996]) give a first level of standardisation of the textual meta-information.

### 3.1.2 *Structured data*

The XML language specified by the world wide web consortium (W3C) [W3C, 2008] gives a structure to textual data. As a markup language[1] textual data are enclosed into tags which can be given attributes. Different dialects of XML can be specified using a document type (Doctype) or an XML schema. XML heavily relies on the notion of uniform resource identifier (URI). In XML, elements of description can be uniquely defined by the URI. For instance in the listing 1, the tag playlist encapsulates a tracklist which is a list of track elements. The playlist has attributes like version and the namespace xmlns which refers to an URI where to find the a specification of the dialect which is used in the document. The elements of the XSPF playlist of our example are music tracks which can be accessed at a remote [2] or on a local path of the filesystem accessible to the application. XSPF is a very simple and basic dialect which is only intended to describe playlists. Much more complex metadata formats have been designed in order to cover a wider range of use cases. The MPEG consortium has worked on the definition of exhaustive descriptors for the media content MPEG-7, and for the usage of media content MPEG-21 and the user[3]. Specific applications lead to the development of subset dialects like TV-Anytime, and EBU for electronic program guide (EPG) systems. Some standards like RadioVIS[4] are designed to display text metadata synchronized with audio or video data.

Listing 1: A playlist with the XSPF standardised XML dialect

```
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/">
  <trackList>
    <track>
      <title>The other Way</title>
      <location>http://previews.7digital.com/clips/34/2197882.clip.
          mp3</location>
    </track>
    <track>
      <title>Photograph</title>
      <location>http://previews.7digital.com/clips/34/163889.clip.
          mp3</location>
    </track>
  </trackList>
</playlist>
```

Most of the time XML is stored in a text format because it is very convenient for a human operator to edit it. However, this format has some limitation for the exchange and processing of information: Text data must be parsed and mapped to runtime objects (marshalling) when being processed; it is suboptimal to store some types of values (like floats, ints, *etc.*) in text. A solution can be to use compression

---

[1] markups languages have been standardised by the SGML specifications [Iso, 1986]

[2] an uniform resource location (URL) is the location of a resource, if this location is unique, then it can be used as an URI

[3] The purpose of the MPEG-7 and MPEG-21 standards are exhaustively presented by Chariglione who very actively supported the MPEG consortium for the ISO standardisation, on his website http://mpeg.chiariglione.org/, especially on http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm for MPEG-7 and on http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm for MPEG-21

[4] RadioVIS is a part of the RadioDNS standard

formats (like ZIP) to store metadata, but it would add overhead to the processing without solving the parsing problem and would make random access cumbersome. Another solution is to develop a binary format for XML, which enables the fast processing and storage of XML data. Unfortunately the standardisation effort has not lead yet to a wide spread industry adoption and the binary format for xml (BiM) standard of the MPEG consortium [ISO, 2006] is coexisting with proprietary solutions.

### 3.1.3 *Linked data and semantic web*

The increasing amount of online metadata and the growing support of standards has lead more recently to the apparition of two trends. On the one hand, metadata sources are linked between them, for instance the unique identification of a location related to an information hosted by a podcast provider, can be linked to the description of this location or to photographs with the same location. On the other hand, the elements which are used to describe assets (*e.g.* the tags and the attributes in XML) can be given logical rules to describe them. The combination of those two trends is called the semantic web and should permit a broad range of new media access scenarios[5]. The W3C has produced a framework called resource description framework (RDF) which can capture different sets of logical statements called description logics. Description logics are subsets of first order logic. RDF expresses knowledge in the form of triples: Subject, Predicate, Object, the predicate being a logical quantifier which gives the meaning of the link betwenn the subject and the predicate. Based on RDF another standard, ontology web language (OWL) allows to specify knowledge on a specific domain. The listing 2 illustrates semantic metadata about the band *Shiva in exile* aggregated by the DBTune and using partially the Music Ontology developed by Raimond [2008]. For instance the subject `mo:`MusicArtist shiva_in_exile which is an individual instantiation the `MusicArtist` class of the music ontology[6], has the predicate of the FOAF[7] property `based_near` with the object `Germany` which is an instantiation of a class of the DBpedia ontology[8]. The biography of the artist is given with the <bio:olb> tag, and the address where to find the music (Magnature[9], distributing independent artists). In this example, three different ontologies are used to make an assertion about the geographic origin of an artist. Therefore, a model storing all this structured information can be queried against an infinite types of constraints. For instance, we could ask the DBTune to list all the composers which are located in Germany, using over types of information on them, we could refine the search.

Listing 2: DBTune metainformation about the band Shiva in Exile

```
<rdf:RDF>
```

---

[5] The first introduction to the notion of semantic web has been formulated by Tim Bernard Lee [Lee, 1999]

[6] The music ontology gives an semantic descriptions to the main concepts about describing music: http://musicontology.com/

[7] friend of a friend (FOAF) or friend of a friend ontology is an ontology for the description of people and their relationships: http://xmlns.com/foaf/spec/

[8] The DBpedia is a project aiming at analysing and structuring the content of Wikipedia: http://dbpedia.org/

[9] www.magnatune.com

```
 <rdf:Description rdf:about="http://dbtune.org/magnatune/album/
     shiva-ethnic">
  <foaf:maker rdf:resource="http://dbtune.org/magnatune/artist/
      shiva_in_exile"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://dbtune.org/magnatune/album/
     shiva-nour">
  <foaf:maker rdf:resource="http://dbtune.org/magnatune/artist/
      shiva_in_exile"/>
 </rdf:Description>
 <mo:MusicArtist rdf:about="http://dbtune.org/magnatune/artist/
     shiva_in_exile">
  <bio:olb rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A
       journey from middle to far east echoed by simple world
      rhythms and resounding through an ambient backdrop ...</
      bio:olb>
  <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#
      string">Gothic Arab/Indian World Music</dc:description>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string
      ">Shiva in Exile</rdfs:label>
  <foaf:based_near rdf:resource="http://dbpedia.org/resource/
      Germany"/>
  <foaf:homepage rdf:resource="http://magnatune.com/artists/
      shiva_in_exile"/>
  <foaf:img rdf:resource="http://magnatune.com//artists/img/
      shiva_in_exile.jpg"/>
  <foaf:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Shiva in Exile</foaf:name>
 </mo:MusicArtist>
 <rdf:Description rdf:about="http://dbtune.org/magnatune/
     performance/2740397">
  <mo:performer rdf:resource="http://dbtune.org/magnatune/artist/
      shiva_in_exile"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://dbtune.org/magnatune/
     performance/2740398">
  <mo:performer rdf:resource="http://dbtune.org/magnatune/artist/
      shiva_in_exile"/>
 </rdf:Description>
</rdf:RDF>
```

### 3.1.4 *Embedded metadata currently used in vehicles*

The metadata can be embedded with the media assets themselves. The standards theoretically support very complex combination such as the MPEG-4 container which can encapsulate MPEG-7 metadata. The industry is most of the time supporting subsets of the standards, in order to reduce the complexity of the platforms which have to be maintained.

RDS Before digital broadcast was available and EPG systems included in the last generation of headunits, the radio data system (RDS) and RDS plus standards where develop by radio broadcasters in order to inform receivers about the name of the station and about the name of the program currently broadcasted [Quelle and Kusche, 2006].

EPG The electronic program guide format has been developed by digital broadcasters in order to substitute to teletext that was used in the analog world. In Europe the EBU is a standardising authority and had produced TV-Anytime based on MPEG-7. EPG enables end devices manufacturers (TV-sets, IPTV-boxes, *etc.*) to add interactive program guides with menus displaying information about current and coming programs.

ID3 ID3 tags were designed to work with the MP3 container format[10]. In the original version of the specification, ID3 tags could store on 128 bytes the following information: title, artist, album, year, comment, and genre (in a list of 255)[11].

## 3.2 THE DESCRIPTION OF MEDIA ASSETS

Now that we know that the problem of knowledge representation has already been solved. We need to analyse, what are the types of metatada that we need, and how to access them. The space of possible media descriptions is infinite and we will not pretend to exhaustively list all possible sorts[12]. Back to the scenarios that we have defined in section 2, we can restrict our analysis to some specific information types.

### 3.2.1 *Information content description*

Alice (see section 2.1) needs to listen to reports about politics and economics. That means she needs to be able to select a topic and a format.

#### 3.2.1.1 *The types of information*

INFORMATION TOPIC The topic of a program describes the general subjects which are tackled: politics, economics, sports, *etc.* Moreover, the language in which the program is recorded.

INFORMATION FORMAT The format defines the way the program is recorded/broadcasted. It can be a talk-show, or a live transmission (for sports, for instance), a journalist report, a documentary, *etc.*

#### 3.2.1.2 *Use in the industry*

Currently, the main form of on demand assets (podcast and vodcast) is using the taxonomy defined by iTunes for information topics. This taxonomy is compound of 17 categories and 51 subcategories [Apple, 2011]. Most of this data is structured but not semantic yet (it is not possible to process logical reasoning on them). More complex vocabularies have been designed but are not yet very widespread. Troncy [2010]

---

[10] Actually the container format for MP3 can be either MPEG-1 or MPEG-2 Audio Layer 3 [MPEG, 1993, 1995]

[11] more specifications on ID3 are available at www.id3.org

[12] The TV-Anytime which is based on a subset of MPEG-7 defines alone 13 classification schemes: ActionTypeCS, AtmosphereCS, ContentAlertCS, ContentCommercialCS, ContentCS, FormatCS, HowRelatedCS, IntendedAudienceCS, IntentionCS, LanguageCS, MediaTypeCS, OriginationCS, TVARoleCS and RoleCS.

investigated a semantization of NewsML[13] in order to demonstrate that semantic queries could be performed against semantic datasets of news information in an interactive manner. Troncy demonstrated the importance of a time model when describing information but acknowledges there is still much to do in terms of time reasoning, and also in terms of dynamic acquisition of semantic data.

### 3.2.2 *Music content description*

When Pachet and Cazaly [2000] and Aucouturier and Pachet [2003] started to investigate on music taxonomies, they first looked at the classification used by the music industry and came to the conclusion that there were no unified taxonomy of music. As always, when trying to classify objects, it is necessary to identify what is the purpose of this classification. Music retailers want to sell CDs that they expose in their music stores, therefore they use a rather simple classification by genre probably because it is easier to maintain their shelving so. Online retailers want customers to find music content on their websites by clicking on tabs, therefore they use more refined classifications, *etc.* We will focus in this section only on music specific classification, and in the next section we will extend to music non-specific descriptions. In our scenario, Alice wants to select music by genre but she also defines other types of qualifications for her choice like 'relaxing'. Moreover, she might want to select specific tracks or artists to show her findings to her colleagues.

#### 3.2.2.1 *Type of music classification information*

MUSIC PERFORMANCE INFORMATION Everyone knows the name, the tracks and even the albums of its favorite band. The information like the composer, the interpret, the title of the track, the title of the album are therefore very valuable when describing music content. Moreover, the music age is also another important aspect

MUSIC GENRE The musical genre is a high-level concept which helps to group music artworks which have commonly accepted similarities in their composition. The notion of genre for music is widely studied but never came to a consensus [Aucouturier and Pachet, 2003, Hahn, 2010]. In his PhD Thesis on automatic genre classification for music Guaus [2009] emphasizes on the difference between genre which is something commonly accepted and style which is more specific. Pachet and Cazaly [2000] demonstrate that not only if the meaning of a genre can be commonly accepted, variations can occur according to the cultural background. Moreover, he adds that the hierarchical dependencies between the genre are various (geography dependency, genealogy dependency, *etc.*).

MUSIC MOOD Music belongs to entertainment, and thus has an influence on the psychology of whom is listening to it. The valence arousal scale introduced by Russell *et al.* [1989], Russell [1980] is currently one of the most used for music mood taxonomies [Schuller *et al.*, 2010]. It consists in a two dimension scale: (1) the valence, which indicates

---

[13] NewsML is another vocabulary for news developed by the international press communication council (IPTC)

the pleasure and displeasure and (2) the arousal, which is a notion of emotional intensity.

MELODIC AND RHYTHMIC, INSTRUMENTAL Other types of music classification can be used. The mode for instance (major, minor), the rhythm (slow to fast), the type of instrument or the presence of human voice are music descriptions which can be used to classify and select music assets.

#### 3.2.2.2 *Syntactic and semantic representation*

Research projects like the MX ontology [Ferrara *et al.*, 2006], the music ontology [Raimond, 2008] have demonstrated the potential of semantic representation of music metadata, by a lot of scenarios. The MPEG-7 offers already a large amount of descriptors for music with its Audio Framework which covers mainly low level descriptors. Moreover, it is possible to create new classification schemes within MPEG-7 (for instance, it is possible to create a specific classification scheme, for a genre vocabulary). Different ontologies of MPEG-7 Classification Schemes have been produced and the semantization of MPEG-7 is still a research topic. In the industry some metadata providers like Gracenote[14] or The All Music Guide[15] or The Echonest provide all the aforementioned types of metadata in order to enrich media databases. Raimond has shown that it is possible to map dynamically structured data from the Echonest for processing within the music ontology.

## 3.3 DESCRIBING USERS AND USAGES

Metadata can be used not only to describe content but also to describe users. The tendency to store online more or less formalised definition of user characteristics including consumption preferences has boomed with the advent of social networks on the internet [Piotet, 2008]. In our scenario, the user profile of Alice, regarding her music or information preferences need to be made persistent.

### 3.3.1 *User preferences*

The MPEG-7 description schemes gives a structured vocabulary and syntax to define user preferences using the XML schema definition (XSD) language (see figure 14). For instance, if a MPEG-7 `ClassificationScheme` defines the `term` jazz as a genre. It is possible to define a `UsagePreference` object referring to this `term` of the `ClassificationScheme` and to give to this object a `preferenceValue`. Those are syntactic considerations and have been implemented for example by Wang *et al.* [2004] in a news agent personalise. One of the more elaborate type of user preference reasoning based on metadata has been proposed in the AVATAR project by Blanco-Fernández *et al.* [2008] and it is using a semantic model. Semantic models use description logics to build complex rules for the reasoning. They can infer statements, for instance if for each music track, the list of instruments is stored in a semantic description and that a user formulates a preference like “ all music tracks with string instruments”. If the ontology describes string instruments as instruments

---

[14] www.gracenote.com
[15] www.allmusic.com

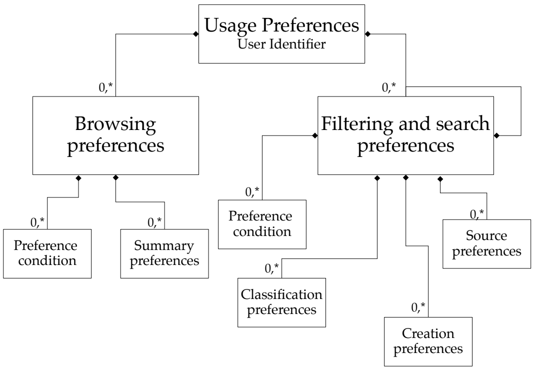


Figure 14: UML presentation of MPEG-7 UserPreference description schemes elements [Manjunath *et al.*, 2002, chap. 11]

producing sounds through vibrating strings and that for each instrument in the ontology the type of sound production (bowing strings, plucking strings, striking string) is given, then an inference engine can filter all the tracks which meet the preference of the user. The value given to a preference is somewhat difficult to formalize. [Manjunath *et al.*, 2002, chap. 11] suggests for the MPEG-7 `DescriptionScheme` to give values comprised between $-100$ and $100$, a negative value indicates that the user does not like an item or a sort of item and a positive value indicates the opposite, such that when combining different preferences, likes and dislike can compensate. The user preference dilemma is very often considered as a probabilistic problem since it is used to filter content and especially to process recommendation. We will see in section 3.5 that it is manyfold issue.

### 3.3.2 *User history*

In order to make recommendations and to populate the user profile, the usage of history data can be put to good use. The MPEG-7 `DescriptionScheme` for `UsageHistory` (see fig. 15) relies on different `ActionTypes` which can be defined using a `ClassificationScheme`. This history captures, user events and with time values (date, or length). However, MPEG-7 does not provide any language to store contextualised history. Weiß [2009] introduced in a her doctoral thesis an XML language extending MPEG-21 and MPEG-7 with contextual features in order to express contextualised preferences on a syntactic base (see section 6.4.1).

### 3.3.3 *User communities and idiosyncratic tagging*

As we said formerly, the social networks that we have presented in section 2.4.3 are today the biggest aggregators of user profiles in the infotainment domain. In our scenario, Alice likes to share her music tastes with her friends when in the vehicle, but she can also do it online. The service Last.FM has developed a protocol called scrobbling to monitor the music behavior of its users. It consists in collecting through the multimedia client, for a given user listening to music the name of the track and of the artist which get consummed. Social

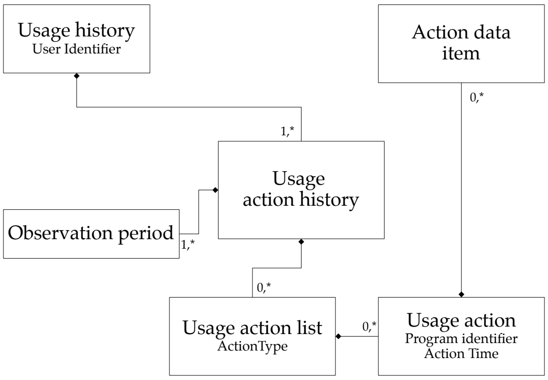


Figure 15: UML presentation of MPEG-7 UserHistory description schemes elements [Manjunath *et al.*, 2002, chap. 11]

networks also propose to their users to store unstructured metadata as tags. For instance Last.FM users have tagged the track *Smooth Operator* interpreted by Sade, with the following words: 80s, female vocalist, jazz, soul, chillout, atmospheric, *etc.*. Those are called unstructured since tags can represent various type of information such as the music age, the type of performance, the genre, the mood or the purpose of listening to this kind of music. Some authors claim that this common knowledge creates a sort of network intelligence[16]. Cunningham *et al.* [2004] have established that users tend to organise their music collections according to certain playback purpose. Social networks give the possibility to compare what the people share in common, in the way they use their music. In short we can summarise the role of the social networks in the metadata chain as following.

- socialising functionality: people can compare their music profiles with other people (information sharing), in order to make the acquaintance of people sharing common interests
- discovery functionality: people can discover new music, using profiling and similarity recommendation
- socialised functionality: groups of people (local community) can be offered a dedicated program

## 3.4 THE METAKNOWLEDGE WORKFLOWS

In the whole value chain of the media production and consumption, metadata are produced from different entities. This has consequences in terms of information quality, availability and processing which have to be taken into account in our on-demand architecture.

### 3.4.1 *Metadata extraction/creation*

Metadata is not always available, and this is a significant aspect when designing a vehicle application. There is a functional link and a technical link between the different sort of metadata creation techniques. Very often, in order to identify the media assets which are to be edited

---

[16] Piotet [2008] introduce the notion of ‘intelligence réticulaire’ .

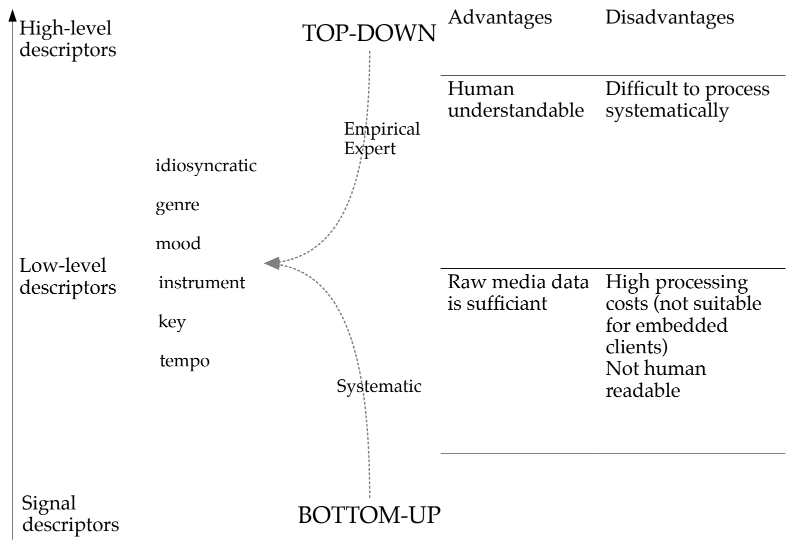


Figure 16: Bottom-up and Top-down creation of metadata

with metadata, it is necessary to use raw signal analysis. The figure 16 summarizes the different types of metadata creation techniques. We can oppose bottom-up to top-down extraction.

### 3.4.1.1 *Bottom-up extraction*

The bottom-up extraction consists in getting metadata features from the raw data (audio and video signal).

MUSIC LOW-LEVEL FEATURES EXTRACTION In the audio-domain, different low-level features (MFCCs or MPEG-7) are extracted from the audio signal [Kim *et al.*, 2005]. Recommendation can be then based on data mining technique [Kastner *et al.*, 2002], [Haitsma and Kalker, 2002]. This step is often processed locally.

INFORMATION TAG EXTRACTION Information extraction from audio/video content are speech and image recognition technical issues.The Voxaleadnews project presented by Law-To *et al.* [2009][17] provides an interface to browse podcasts and vodcasts using extracted semantic descriptions. The text information from the news is transcripted and semantic tags are extracted from it. The prototype is organised in a client-server architecture, and Law-To *et al.* [2009] acknowledge a rather resource demanding speech recognition and keywords extraction which are processed on a server.

### 3.4.1.2 *Top-down creation*

It is also possible to create metadata by other means than the extraction of features, especially using a human knowledge and analysis. This is what we call top-down creation.

---

[17] A demonstrator of the project is available at the following address: http://voxaleadnews.labs.exalead.com

EDITORIAL METADATA The metadata containing about the performance of music tracks (composer, title, *etc.*) or the topic of information are editorially processed information by the editing or publishing entity.

EXPERT METADATA Expert data cover domain specific metadata like the mood or the genre of the music that we have formerly described. The music genome project conceived by Glaser and Westergren consists in representing music tracks as vectors of 400 'genes' [Glaser *et al.*, 2006], each gene being a musical attribute given by an expert who listened to the track. For each track, experts submit assessments of the features to a database. Similarly, the All Music Guide database [18] provides expert information to sort music according to genre, moods, instruments and give editor's recommendations.

FOLKSONOMIES Folksonomies also known as 'wisdom of folks' are composed of aggregated data in social networks: such as tags, playlists, item ratings. Folksonomies must be seen as active datasets, and the monitoring of their evolution (most clicked video of youtube, most rated artist of Last.FM, etc.) contains actually as much information as the data they actually contain. The quality and exactness of the tagging must be kept consistant otherwise it can get rapidly unusable [Peters and Weller, 2008]. From the tagging, rating, playlist-cooccurence information, it is possible to extract so called social filtering recommendations (see section 3.5).

SOCIOLOGICAL STUDIES The idiosyncratic annotation of the user needs in terms of media use is a subject of study for media sociologists. Lesaffre has analysed the way people interact with media indexing and retrieval (MIR) systems in order to find out a user-centered taxonomy of music [Lesaffre *et al.*, 2003, Lesaffre, 2006]. Her study "suggests a five-leveled conceptual framework that involves acoustical, sensorial, perceptual and expressive description levels". She pointed out that when user are looking for a specific track they prefer to use the title and the genre, whereas when they are looking for unknown music they prefer to use genre and rhythm.

### 3.4.2 *Metadata access for mobile clients*

The access to metadata from the point of view of a media client can have noticeable architecture consequences: Do the client need to perform the metadata extraction or can it retrieve them from a service?

#### 3.4.2.1 *Self-extracted metadata*

The client can extract by itself using bottom-up techniques. It has the advantage, that no external service is needed and that the quality of the metadata only relies on the quality of the algorithm. The main drawback is the need for computing resources on the client in order to extract the features. Self-extraction of the metadata is a good solution when data are already locally stored on the client.

---

[18] www.allmusic.com

| Last.FM | | | Echo Nest | | |
|---|---|---|---|---|---|
| average RT | min | max | average RT | min | max |
| 0.82 | 0.44 | 5.41 | 0.75 | 0.2 | 33.0 |

Table 3: Response time (in seconds) of two metadata providers

#### 3.4.2.2 *Metadata co-delivery and separate delivery*

Very often, the metadata is delivered together with the content (*e.g.* music ID3 tags, XML description of podcasts). The quality of such editorial data is most of the time very good but also heavily relies on the commercial strategy of the provider. For instance, a music provider with a little classical music catalogue might want to use a taxonomy of music genres where classical genres are underrepresented. Discrepancies can occur among various providers which use unstandardised or inconsistent thesauri. Some types of high-level metadata are available only through the request to external services. This is possible only under the condition that the media assets can be uniquely identified by the external provider. The disadvantage is that the client needs to perform an additional network access when accessing external services. On the other hand, pure metadata providers such as web crawlers or folksonomies tend to use exhaustive or standard representation of data in order to be compatible with a wide range of media assets providers. The request to external services can introduce latency in mobile applications. We have measured for a total of 1000 queries for each, the response time of Last.FM and the Echo Nest (see table 3). As we can see, they are on average under a second, which is sufficient for a connected application. But the variation can be extremely high which could result in anreactive interface, if metadata are retrieved directly from the vehicle.

#### 3.4.2.3 *Aggregation of metadata: media assets identification*

When metadata come from different sources it can be difficult to aggregate them. The encoding, or the spelling of the textual data can vary a lot [19].

IDENTIFICATION FROM THE RAW SIGNAL It is first possible to try to identify the data from the raw signal directly in order to reconstruct robust metadata. MPEG-7 features [Allamanche, 2001] or MFCC [Berenzweig *et al.*, 2004] features can be extracted from the audio signal in order to compute fingerprints. This solution is convenient if unidentified media (for instance no ID3 tags are used) have to be analysed by the client but is far too resource demanding for a media aggregation process.

HAMMING DISTANCE, AND LEVENSTEIN DISTANCE If metadata from different sources have to be merged, it can be worth to compare distance between text strings. The Hamming distance compares one by one the differences between characters at the same position. It is very easy to compute and has a linear complexity. However, the

---

[19] The Last.FM developers have noticed for instance 100 different ways to write *"Guns N' Roses -Knockin' on Heaven's Door"*: http://blog.last.fm/2008/03/25/fingerprinting-and-metadata-progress-report

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | U | N | S | | A | N | D | | R | O | S | E | S |
| G | U | N | S | | N | | R | O | S | E | S | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 4 | 5 | 6 | 6 | 7 | 8 |

Table 4: Hamming distance between text strings `GUNS AND ROSES` and `GUNS N ROSES`

| | | G | U | N | S | | A | N | D | | R | O | S | E | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| G | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| U | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| N | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| S | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| N | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 2 | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |
| R | 8 | 7 | 6 | 5 | 4 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 4 | 5 | 6 |
| O | 9 | 8 | 7 | 6 | 5 | 4 | 4 | 4 | 4 | 4 | 3 | 2 | 3 | 4 | 5 |
| S | 10 | 9 | 8 | 7 | 6 | 5 | 5 | 5 | 5 | 5 | 4 | 3 | 2 | 3 | 4 |
| E | 11 | 10 | 9 | 8 | 7 | 6 | 6 | 6 | 6 | 6 | 5 | 4 | 3 | 2 | 3 |
| S | 12 | 11 | 10 | 9 | 8 | 7 | 7 | 7 | 7 | 7 | 6 | 5 | 4 | 3 | 2 |

Table 5: Levenshtein distance between text strings `GUNS AND ROSES` and `GUNS N ROSES`

hamming distance is too sensitive and cannot recognise deletions (see table 4). An alternative to the hamming distance is the Levenshtein distance [Levenshtein, 1966] also known as edit distance. It computes the minimal number of insertions and deletions to be done when transforming a string of length $m$ into a string of length $n$ (see an example in table 5. A drawback of the Levenshtein distance is that it is difficult to achieve a better complexity better than $O(m \times n)$.

**SOUNDEX** Another solution to compare textual metadata is to compare the pronunciation. The soundex is a method of syllabic transcription of words into a code system [The National Archives, 2007]. Therefore, words with the same or similar pronunciation can be transcripted with same or similar soundex codes. For instance in `GUNS AND ROSES` is transcripted as `G5253` and `GUNS N ROSES` is transcripted as `G5256`.

### 3.4.3 *Summary of metadata workflows*

To fulfill its multiple personalisation roles (see section2.5) the vehicle as a multimedia client relies on different sources of metacontent (see figure 17). The metadata coming from the new actors in the digital value chain (folksonomy, web crawlers) has to be aggregated as well as metadata coming from the content providers themselves. We will present in chapter 5 an architecture that deals with the problem of metadata latency.

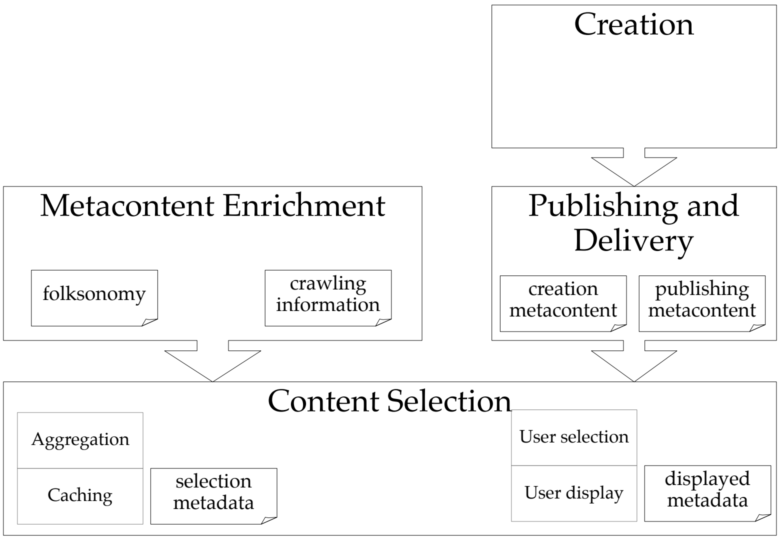


Figure 17: Metadata in the media consumption value chain

## 3.5 RECOMMENDATION OF MEDIA: THE SPECIAL CASE OF MUSIC

We have seen the previous sections the different sorts of metadata which can be associated to infotainment assets and to users. A very illustrative way of seeing the use of metadata is to use them for recommendation purpose, *i.e.* to make suggestions of content. Based on metadata, we can process different kinds of recommendation. The two first sorts are called content-based filtering. Using social information, two other sorts of filtering can be processed: context-based (using tags) and collaborative filtering (using playlist co-occurrence).

### 3.5.1 *Content based recommendation*

"Content-based recommenders provide recommendations by comparing representations of content describing an item to representations of content that interests the user" [Melville and Sindhwani, 2010]. For instance if a user says that he likes listening to jazz music, then a recommender can suggest music tracks which are tagged with the jazz genre. If genres are represented with taxonomic structure or within an ontologies, then the recommender can suggest tracks from related genres like 'acid jazz' or 'classic jazz'.

RECOMMENDATION BASED ON ACCOUSTIC SIMILARITY The recommendation based on accoustic similarity is already a very widely studied topic [Pampalk, 2006, West *et al.*, 2006, Berenzweig *et al.*, 2004] which belongs to the same field of study as music track identification. The Simple Music Generator presented by Pampalk and Gasser [Pampalk and Gasser, 2006] is a proof of concept that can generate music playlists from a local database. Their prototype underlined the necessity to add some variance to the similarity estimation in order to avoid repeating similar songs to often. This variance can be set on the tracks and on the artists.The advantage of accoustic similarity recommendation methods is that it is easy to implement when dealing with local content. No additional textual metadata is necessary since those techniques are only based on a signal processing of the raw audio data, the more features can be extracted the more parameters can be used for the recommendation. However, those techniques do not scale to a mobile aggregation scenario, where data comes from different media sources and the networking and computing resources are limited.

RECOMMENDATION BASED ON EXPERT DATA Instead of relying on signal processing, the recommendation can be based on a subjective analysis of the content. Depending on the genre, the type of performer, orchestration, *etc.* the distance between music tracks can be computed using a distance function (see formal definition in appendix *p.*215.[20] Expert recommendation, can be, as we have formerly noticed, independent from the provider and from the raw data signal. They do not require extra processing. That makes them good candidates for a mobile application. The PATS prototype developed by Pauws and Eggen [2002] has been deployed on a Philips Pronto remote control connected to a server.

### 3.5.2 *Collaborative filtering*

The two previous recommendation techniques make use of music-related features (content-based recommendation). The online social networks (see section 3.3.3) now provide a new way to rate content using input from the user themselves, and thereby provide so called collaborative filtering (CF). Indeed it is possible to represent for each user the preferences for each item as in table 6. That is to say that given a set of users $\mathcal{U} = \{u_1, \ldots, u_n\}$ who have rated items among a set of items $\mathcal{I} = \{i_1, \ldots, i_m\}$ with some rating value $r$ in $\mathcal{R}$. We can note that users (and it is in practice what happens) do not rate all the items but only subsets of $\mathcal{I}$. Given an active user $u_a$, who has rated the items of $\mathcal{I}_{u_a} \subset \mathcal{I}$, the role of collaborative filtering can be (1) to process a prediction of the rating of a given item $i_k$ or (2) to give a list of items which would maximize the ratings of user $u_a$ [Sarwar *et al.*, 2001]. Hofmann [2004] formalises the prediction problem as a probability problem.

1. *forced prediction*: the probability that the user $u_a$ rates item $i_k$ with value $r$: $P(r|u_a, i_k)$. This probability supposes that the user is explicitly asked to rate the item.
2. *free prediction*: $P(r, i_k|u_a)$. This probability supposes that the user can chose both the item and the value.

The CF filtering can be either based on relationships between items, between users or both.

ITEM BASED CF FILTERING Item based filtering consists in making a prediction based on the columns of the table. That is to say, when predicting the rating for item $i_k$, finding similar items from $\mathcal{I}$ which have already been rated by the user. Two items $i_k$ and $i_j$ are similar if they get similar ratings from all the users. The Pearson correlation is

---

[20] The Pandora[21] is a commercial example based on the music genome project.

Items

| | | $i_1$ | $i_2$ | … | $i_k$ | … | $i_m$ |
|---|---|---|---|---|---|---|---|
| | $u_1$ | 1 | 15 | | | | 3 |
| | $u_2$ | | 4 | | | | |
| | ⋮ | | | | | | |
| Users | $u_p$ | | 4 | | | | 3 |
| | ⋮ | | | | | | |
| | $u_n$ | 3 | | | | | 6 |

Table 6: User item preferences

widely used as a means to estimate similarity in CF algorithms [Sarwar *et al.*, 2001]:

$$\rho_{i_k,i_j} = \frac{\sigma i_k, i_j}{\sigma_{i_k} \dot{\sigma}_{i_j}} \tag{3.1}$$

Where $\sigma i_k, i_j$ is the correlation between $i_k$ and $i_j$ . Different types of similarity functions are explained in Appendix 215.

USER-BASED CF FILTERING Instead of comparing the columns, one can try to compare the users *i.e.*, to find a set of users which are similar to $u_a$ and then to affect a weight to their ratings to user $u_a$. User-based algorithm are basically based on neighborhood search, and are known to have poor online performance [Melville and Sindhwani, 2010].

HYBRID FILTERING, MODEL-BASED FILTERING Finally, hybrid methods try to use relationship both on items and on users. Probabilistic approaches like Hofmann's model [Hofmann, 2004] introduce latent variables in the formulation of the prediction in order to process dimension reduction on both the item side and the user side. More recently, hypegraphs models based not only on user ratings but also on content-based similarity have tried to give an holistic approach to recommendation with very impressive results [Bu *et al.*, 2010].

### 3.5.3 *Preference of several users*

Finally, in a vehicle, it happens quite often that several people are sitting and want to come to an agreement about what is to be listened to. A first rather simple possibility is to filter only items which have negative preferences and to let everything else pass through [Chao *et al.*, 2005]. The recommendation of a group of people has been tackled by O'Connor *et al.* [2001] within the MusicLens project. O'Connor *et al.* [2001] suggest mainly two ways to process group recommendation:

1. Create a pseudo-user which characteristics represents group's tastes. O'Connor points out that this technique has some drawbacks: People who have given a lot of different preferences might be privileged in the pseud-user's characteristics. Moreover, the output of the pseudo-user's recommendation might be difficult to predict and to understand for the users.

2. Get recommendations for all the users, and merge those recommendations. If this technique is easier to understand for the user, it might not always find optimal solutions.

However, those approaches tend to forget that a playlist has a time dimension, that is to say that instead of trying to please all the user together at a given time, it is possible to give some users in the group more importance at a given moment and then to reward the others for their patience. Baccigalupo and Plaza [2007] name this principle poolcasting for the so called 'social' web radio that they have developed.

## 3.6 CONCLUSION

Metadata are playing a very important role in the consumption process of digital content. They give us information about the media assets and if they are properly used in the distribution architecture, they can help save media processing, and network access. One should avoid bottom-up extraction of descriptive metadata in connected vehicle scenarios since it involves much computation resources. The use of top-down created metadata from expert knowledge or from folksonomies is to be preferred for the recommendation of online content. When aggregating metadata from different providers, it is necessary to use techniques to assure the exactness and the consistency of the metadata (different taxonomies, different spellings, different languages).

# 4

# ON-DEMAND MEDIA DELIVERY: THE PERSONAL RADIO ARCHITECTURE

> *" The purpose of this Act is to help ensure the continued leadership of the United States in high-performance computing and its applications by:*
> *(1) expanding Federal support for research, development, and application of high-performance computing in order to:*
> *(A) establish a high-capacity and high-speed National Research and Education Network;*
> [. . .]
> *(E) promote the more rapid development and wider distribution of computing software tools and applications software;*
> [. . .]
> *(F) accelerate the development of computing systems and subsystems;"*
>
> — Senator Albert Gore Jr., *High Performance Computing and Communication Act, 1991*

This chapter defines the media delivery platform called personal radio and introduces a layered web-architecture for the consumption of on-demand content. For this, we first analyse the different sort of mediadelivery in a vehicle in the functional requirements of personalised media and of the technical requirements of a vehicle. We first explain why among others a client/server based on a IP network is the best solution to achieve our goal. Afterwards, we describe the architecture of a rich vehicle client and its integration with existing infotainment services with a remote abstraction layer.

## Contents

## 4.1 MOBILE MEDIA CONSUMPTION

Consuming media in a vehicle needs to be an entertaining activity. As we will see, in this chapter and the two followings, the functional requirements which derivate from the use cases, highly influence the technical specifications of the architectural design of the solution. However, we also need to cope the functional requirements with the technical constraints of a vehicle embedded software (see section 2.6)

### 4.1.1 *Common technical functionalities*

All mobile multimedia devices share common functionalities no matter what technical solution they rely on:

FIND CONTENT The finding of content includes the resolution of an indexed source. This plays an import role in the personalisation. As we will see, there are various architecture solutions to find content.

SELECT CONTENT This entails, the formulation of the constraint defining the source which is an important functionality in a vehicle as we will see in the chapter 5. The selection of content, is not directly bound to the architecture. In the current chapter we will assume that the vehicle media system already has a user interface which can be programmed, and we will reuse for our prototype the RemoteHMI as presented in section 2.9.3.1.

RENDER CONTENT The rendering of the content consists in the decoding of the media which always performs at the reception.

### 4.1.2 *Media delivery paradigms*

The architecture of a media delivery system is heavily dependent on the way the media are consumed. The goal of any delivery architecture is to make a junction between a resource and a user and we can distinguish basically two ways to do it.

PUSHED MEDIA ACCESS In this paradigm, the media is sent from the source to the receiver or client as soon as it has been produced. This paradigms covers all sorts of so called 'live' media, which deliver synchronously the content: live shows, sport events, live news, *etc.*. The media which are scheduled and sent to the receivers based on editorial decisions (radio and TV programs) on which the user has no influence. As we will see the broadcasting architecture are especially well suited for this paradigm.

PULLED MEDIA ACCESS The pulled media-access, also known as 'on-demand' media access functions asynchronously. The media is first produced and stored and the user can access it when he wants to consume it. Even if editorial selections can be proposed to the user, this paradigms enable a user-centered selection. We will in the following of

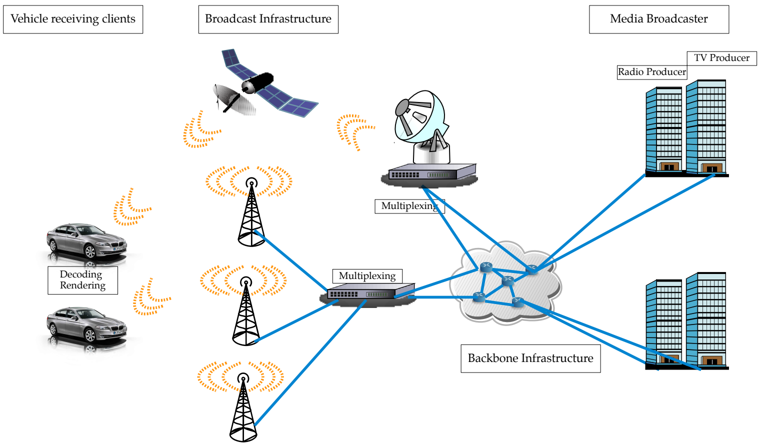


Figure 18: Media broadcast infrastructure

this thesis focus especially on this media access paradigm since it has the biggest personalisation potential.

## 4.2 THE DIFFERENT MOBILE MEDIA DELIVERY ARCHITECTURES

We will detail in this section three major types of media delivery architectures.

### 4.2.1 *The broadcast architecture: one to all*

Historically, the broadcast architecture has played a big role in the spread of media. As soon as the transistor allowed the miniaturisation of radio tuners, most drivers where able to listen to radio programs while driving. For a long time only analog broadcast was available in frequency and amplitude modulation, and more recently digital broadcast has been standardized (T-DMB, DAB, DVB). The broadcast architecture is a one to all architecture (see figure 18). It consists in sending the same signal from a terrestrial antenna (resp. from a satellite antenna) to a an unknow number of users in a given region.

The broadcast has the following advantages: mass production and intensive standardization have reached a price to quality ratio that makes analog broadcast difficult to be replaced in low-end devices. The concept of one to all scales perfectly to urban environments with a concentration of a lots of receptors since adding new receptors does not reduce the performance of the system, and other clients are not affected. Furthermore, passive reception has low energy requirements.

An architectural disadvantage of the one to all architecture, is that everyone can receive the signal and extract the media from it. In order to enable business models which charge the users consuming specific programs, broadcasters encrypt the signal which has to be decrypted by the client. Broadcasters distribute to their customers tamper resistent receivers which make difficult to find out the description key without damaging the receiver.

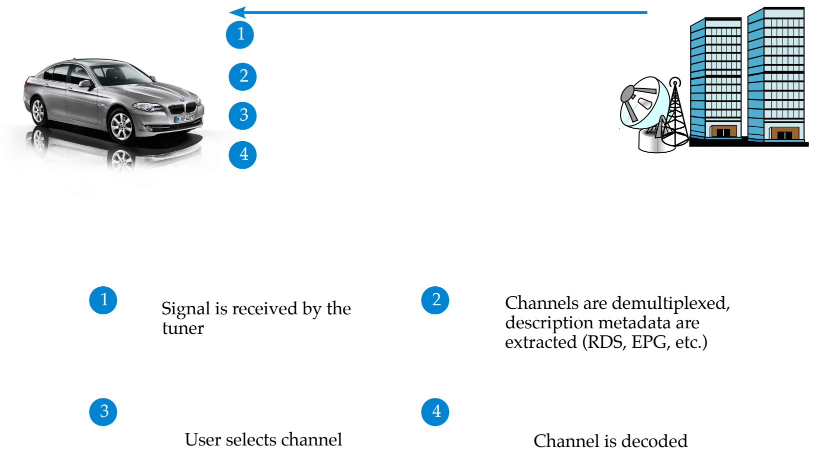


Figure 19: Vehicle media resolution and consumption sequence in a broadcast infrastructure

As a matter of fact, in a broadcast architecture, the operation of selection of the media is completely left to the client (see figure 19), the network only assures the delivery of the same content to all the users. It has an advantage, the system is very reactive since all information are available on the client. A drawback is that the interactions are very limited since there is no possibility for the client to send feedback to the emitter and the amount of information which can be interactively selected is limited to the capacity of the client. The user can display menus and informations related to a given program (like Electronic Program Guide (EPG)), but it supposes that this information has been sent along with the program which is a very strong limitation. As a matter of fact, broadcast architecture are not suitable at all for the on-demand content paradigm and the pull scheduling, they are best suited for push scheduling of live-media.

### 4.2.2 *The routed architecture: one to one and one to many*

Since the beginning of the 21th century, the internet has spread with an incredible speed to mobile devices [Biffen, 2008]. Since 2007, manufacturers started equipping their vehicle with embedded EDGE or UMTS connections offering the possibility to access the richness of web content.

As we have mentioned previously both data and metadata need to be exchanged between the client and the server. The HTTP protocol is well suited for the for this [Kazasis *et al.*, 2003].

The web infrastructure can be summarized as following (see figure 20: Servers containing multimedia data are connected to a routed network, they can receive requests from clients and send them data. The clients are connected to the internet through a bidirectional network. They are given an IP address according to a specific protocol (DHCP), when connecting to the network. Servers are also given an IP address which can be resolved using their URL and a DNS server.

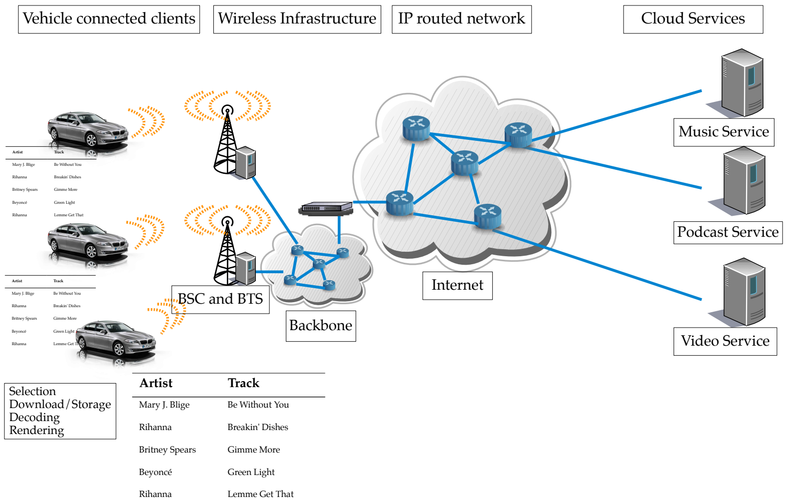


Figure 20: The web infrastructure

The representational state transfer described by Fielding [2000] is currently the most used architecture style for the design of interactive web services. Based on the HTTP 1.1 standard [Fielding *et al.*, 1999], Fielding introduced a clear separation of concerns between servers and clients. The clients handle the user interface and the user states whilst the server handle the data storage. No client context is stored on the server between requests. However, according to Fielding specifications, servers can be stateful, provided that the state of the server can be known by the client when it requests it.

Regarding the consumption of media in a vehicle, the architecture flow can be depicted as in figure 21. In comparison with the flow depicted in 19, the client has the possibility to communicate with the entity delivering the resource. This enables the following functionalities:

- synchronous search and asynchronous delivery of a specific resource (on-demand content and queueing)
- adaptation of the resource (codec adaptation, downscaling, cropping, etc.)
- user relevance feedback

Since the client and the server can exchange information, the complexity of the different entities is bigger than in a broadcasted scenarios and interfaces are more complex.

In order to refine his definition of network-based hierarchical styles, Fielding introduced the concepts of layer and the concept of cache. A layered system "is organized hierarchically, each layer providing services to the layer above it and using services of the layer below it" [Garlan and Perry, 1995]. A cache consists in "a replication of an individual request such that it maybe reused by later requests". [1]

---

[1] The combination of different components in a client/server architecture is discussed by Fielding for the following possibilities : Client-Server, Layered System and Layered-Client-Server, Client-Stateless-Server, Client-Cache-Stateless-Server and Layered-Client-Cache-Stateless-Server [Fielding, 2000].

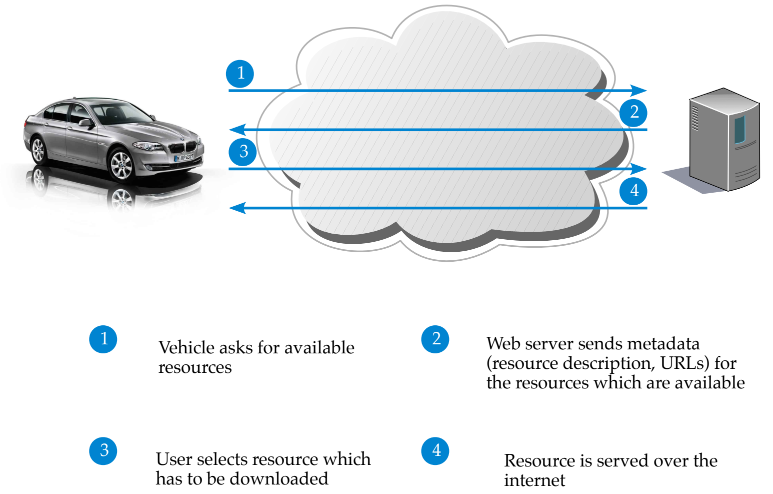


Figure 21: Vehicle media resolution and consumption sequence in a web infrastructure

We will describe in section 4.3 how we designed the personal radio prototype around a client-stateless-server architecture style, moreover, we will in 5.4 present how the introduction of caching in the architecture can be used in order to improve the reactivity of the system and enables new functionalities.

#### 4.2.2.1 *Unicast delivery and media streaming*

In order to propose a better user experience when accessing a specific resource, the technique of media streaming is often used. The main idea is to start rendering a resource before the transfer is finished. We will in the following consider sound media (*i.e* music). Sound medias are encoded with a fixed or variable bitrate (quantity of information per second to encode the media). [2] There are basically two possibilites:

- the network throughput bitrate is bigger or equal to the encoding rate: The media can be streamed and the client can start the playback right after the reception of the first bits of data;
- the network throughput bitrate is smaller than the encoding rate: The media cannot be rendered directly and need to be buffered.

Even if the bitrate is sufficient, interuption can occur during the transfert, especially on mobile clients. This is the reason why buffering is always used. Today the use of non-TCP based streaming protocols for unicast like RTP has dramatically decreased since they are known to be selfish (they have no congestion control algorithm) and compete unfairly with TCP traffic [Lee and Chung, 2008]. Since we target a personalised use of content, we will not deal in the following with multicast IPTV streaming systems.

---

[2] For instance, one second of music encoded with HE-AAC at 64 kbit/s occupies 64 kbit.

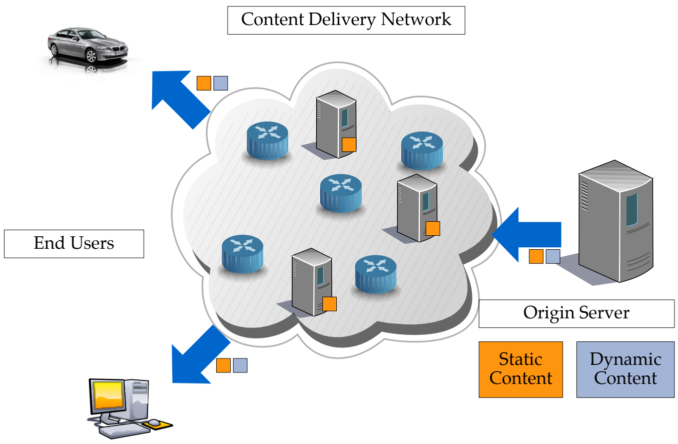


Figure 22: Content delivery network

#### 4.2.2.2 *Content delivery networks*

In order to speed-up the delivery of media content spread over the internet, the so-called CDN (Content Delivery Networks) are networks of servers which are placed at strategic points, sometimes on different backbones in order to maximise the bitrate available for the clients and to reduce the load of the origin server. Content is replicated between different servers or farm of servers caching static information like image, audio or video files. Request routing and load-balancing techniques are then used in order to deliver the media to the client as fast as possible (see figure 22). The advantages of CDNs are [Technofriends, 2008]: a latency reduction (the servers are placed nearer to the clients), a high scalability (the web application can without change on its architecture, scale up to loads of clients), a high availability (clients have less chances do experience service outage) and an improved offload (the design and maintenance of the origin server can only focus on the business logic of the application).

HTTP is a transfer protocol that enables on top of the TCP/IP stack the request and reception of data which are sent by a server to a client.

#### 4.2.2.3 *Case study: Rhapsody Direct and Rhapsody Web Services*

Rhapsody[3] is an US-American music web service launched by Real Networks in 2001. Rhapsody provides a wide range of digital access to music, including the streaming of personalized radios, and the download of purchased content.

FUNCTIONAL ASPECTS OF THE SERVICE Rhapsody provides the possibility to consume music based on an on-demand architecture and is using both a subscription business model and a pay per download business model. A Rhapsody user can search for content using keywords such as artist names, track names, select music genres or just start a 'channel' which is a radio with a specific genre or similar to a

---

[3] www.rhapsody.com

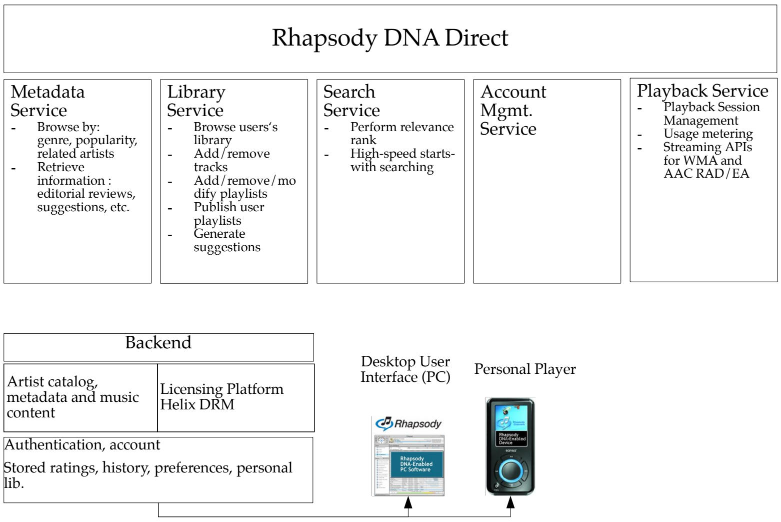


Figure 23: Rhapsody end to end ubiquitous music service distribution (source Rhapsody, 2008)

given artist. Rhapsody claims to have one of the widest catalogues The user can also save playlists with the music he likes in a personal library and navigate through his history. The user can also purchase music, so that he can transfer it to other devices. When the subscription of the user expires, the user cannot listen to the content until he redeems his subscription-fee. User can share their Rhapsody account on different devices, the playlists being automatically synchronised with the remote profile.

INFRASTRUCTURE AND IMPLEMENTATION Rhapsody users can access to the service platform with a specific client which has been developed for computer desktops. Applications for mobile devices like iPhones and Androids and Blackberries are also available. Rhapsody also distributes SOAP API to third Party developers who want to code clients for other platforms and a REST-URL based API for developers who want to integrate Rhapsody on their website. The SOAP API provides all the functionalities of Rhapsody: track and artist search, track streaming, access to user profile.

When a track is selected within Rhapsody, the client has to authenticate in order to retrieve a URL where he can stream the audio content. This URL is uniquely created for this client and automatically expires after being used.

INTEGRABILITY IN A VEHICLE MOBILE CLIENT The SOAP API of Rhapsody allows to implement a client in any language compatible with the vehicle platform. SOAP is known to introduce a certain overhead in the software architecture [Pautasso, 2008a,b] and new sorts of web services based on lighter restfull patterns such as JSON are now preferred. However, the centralized architecture of Rhapsody is an advantage when deploying the application on a mobile client like a vehicle. Rhapsody defines a unified interface to access the user informations and to retrieve the content. The streaming retrieval of

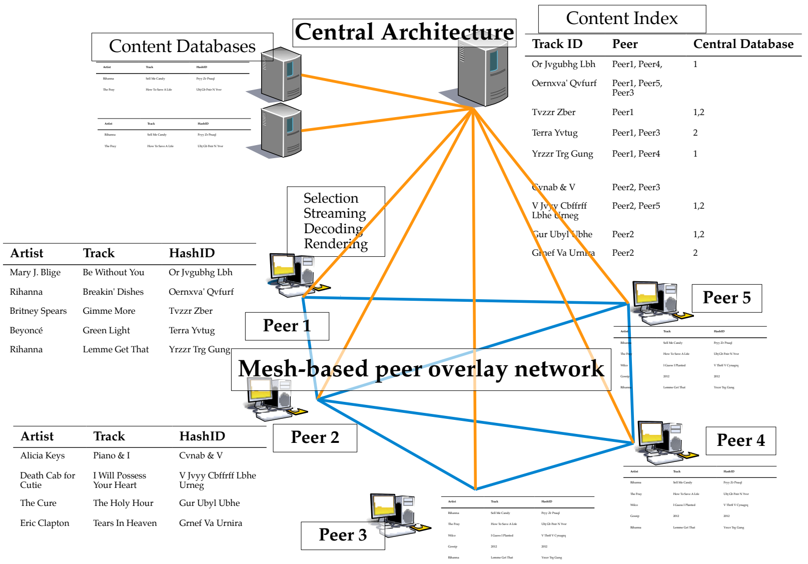


Figure 24: A peer to peer network with a central index mixed with a central storage of files

the media content, combined with an efficient delivery infrastructure offers a scalable approach for a network of vehicles. Moreover, this retrieval technique on a transport protocol, allows vehicle caching of medias which is more resilient to long interruptions in the network connectivity than a pure synchrone delivery.

### 4.2.3 *Peer to peer architecture: one to others, many to many*

The peer to peer services have gained since the launch of Napster in 1998 a very big popular acceptance for the sharing of multimedia content; unfortunately most of the time for illegal purpose. However their success cannot be only attributed to the fact that people could download free of charge content; peer to peer architectures are actually a very interesting approach to deploy large-scale multimedia delivery services.

Peer to peer architectures use network overlays built on the application layer. The node of the network are called peers and are both suppliers and consumers of resources. The idea of a peer-to-peer network is that peers share their local ressources that they have indexed with other peers and it can be employed for media streaming [Ehn *et al.*, 2007, Vodopivec, 2010]. Their are various sorts of peer-to-peer networks, and various way to classify them. It is possible to differentiate them according to the structure of the overlay network; tree-based networks keep a hierarchical structure of the nodes while mesh-based networks connect peers without using any pattern. The purpose of this section is not to give an exhaustive overview of peer-to-peer systems rather to analyze how it would be likely to use it in our on-demand mobile architecture.

#### 4.2.3.1 *Media query over a peer-to-peer architecture*

In peer-to-peer networks, finding a media is based on the matching of the hash of a query string (free-text-search) in a lookup table.

Since resources are distributed over many peers, the media query can be processed in different ways. Either there is a central index (or a tracker) which stores a list of the peers which can supply a given ressource or the index is split between the different peers. In the first solution, anytime a new peer logs in to the network, it sends (or synchronizes) the state of its local content to the central server, when a peer logs out, the server removes it as a potential supplier from its lookup table. This solution is very convenient for services which target a low-latency for the media queries. It has the disadvantage to need a lot of actualization of the centralized data, in a context where peers logs in and out very often and randomly. Another possibility is to spread the information concerning the mapping between resources and peers among the peers themselves. Current peer-to-peer networks use distributed hash tables (DHT) [Wikipedia, 2010a]. DHT make the search of content among a varying among of nodes scalable but also introduces serious security issues [Urdaneta *et al.*, 2009]. The main advantage resides in the capacity of DHT "to handle node arrivals, departures and failures" Wikipedia [2010a].

A third possibility is to flood the query among the other peers until a match is returned by one of the nodes. This last solution, is of course the worst in terms of latency but has the advantage to avoid using centralized servers. As we will see in the nex case study, this is a good architectural solution to provide a scalable music service when used in combination with a structured service.

#### 4.2.3.2 *Case study: Spotify*

Spotify is a swedish music services which has been launched in 2008. Spotify is only based on a subscription model, to a very large catalogue of music

FUNCTIONALITIES OF THE SERVICE The functionality of Spotify are quite similar to those of Rhapsody in terms of music and artist search. Spotify has a Desktop client that users have to install when they want to use their service on their computer. Spotify provides also clients for various mobile platforms and based on their level of subscription, users can synchronise their profiles between their different devices. Spotify does not propose a purchase service, but redirects to a partner provider for music purchase.

INFRASTRUCTURE AND IMPLEMENTATION Even if the functionalities look similar, the music delivery infrastructure is very different from Rhapsody. Spotify relies on a hybrid infrastructure with a peer to peer network and direct download infrastructure. Each client of Spotify is a peer and can stream its music to other peers using the OGG codec. When a peer selects a track for the playback, it has two possibilities, either it can stream the content from another peer or it can stream it from a central content server. Peers can search for other peers both by using a central tracker (peer locator) or by spreading the search to neighbours in a distance of two [Kreitz and Niemela, 2010]. Spotify clients, implements techniques such as peer selection, track caching and TCP congestion control in order to deliver content with a minimal latency and to avoid shutter at playback.

| | broadcast (sender/receiver) | routed networks (client/server) | peer to peer |
|---|---|---|---|
| scalability | ★★★★ | ☆☆★★ | ☆★★★ |
| content personalisation | ☆☆☆★ | ★★★★ | ★★★★ |
| amount of content availabe at a given moment | ☆☆★★ | ☆★★★ | ☆★★★ |
| quality of service (mobile usage) | ★★★★ | ☆★★★ | ☆☆☆★ |
| infrastructure costs/complexity | ☆★★★ | ☆☆★★ | ☆★★★ |
| resource delivering entity complexity<br>(sender, or server) | ☆★★★ | ☆☆★★ | ☆☆☆★ |
| embedded client complexity | ☆★★★ | ☆☆☆★ | ☆☆☆★ |
| genericity of interfaces | ★★★★ | ☆☆☆★ | ☆☆☆☆ |
| pull scheduling | ☆☆☆☆ | ★★★★ | ☆★★★ |
| push scheduling | ★★★★ | ☆☆★★ | ☆☆★★ |

Table 7: Comparison of media delivery architectures

INTEGRABILITY IN A VEHICLE MOBILE CLIENT Spotify also provides an API based on the implementation of a C++ SDK. SDK versions exist for x86, arm and powerpc platforms which are good candidates for an embedded system. However Kreitz and Niemela [2010] noticed that only the official Spotify desktop client support the peer-to-peer functionalities and that it is not yet implemented on mobile clients. Moreover, they underline the overhead of the peer-to-peer protocol which is used to built an overlay network upon the IP/TCP stack and evaluate it to 5.20% of the network resources when streaming music for playback. They did not evaluate the impact of frequent network disconection on the overall performance of peer-to-peer service, but we can guess that it is superior to the previous architecture scheme.

### 4.2.4 *Summary and evaluation of the different architecture styles for the media delivery*

We can summarize the former sections in the table 7. The client/server architecture is the best architecture for our personalisation scenario if we consider the current media delivery technologies.

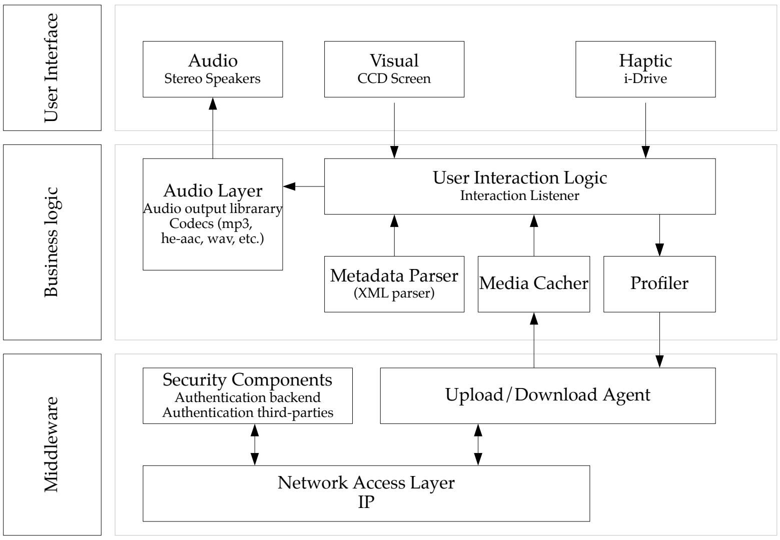


Figure 25: Client technical architecture

## 4.3 PERSONAL RADIO: AN ARCHITECTURE FOR A PERSONALISED MEDIA DELIVERY IN VEHICLES

The personal radio implements an architecture with regards to the comments that we have made in the previous section. It is composed of a layered client-server architecture. Actually in order to avoid developing an interface for each specific media content provider which delivers content, we have decided to introduce a remote abstraction layer that we will further detail. Moreover, a web frontend has been developed for administration purpose: customer management, client profile management, management of user channels.

### 4.3.1 *The client*

The client of personal radio is a rich client application which can be integrated to the software of a vehicle headunit. We have designed it at the beginning of in order to make a demonstrator for certain applications of on-demand media. The core use case being the "play more like this" functionality. The design of the software underlined some technical requirements such as the necessity to be able to playback content when no internet connection is available.

#### 4.3.1.1 *The functional architecture*

We use in the following the notion of functional architecture as given by Broy *et al.* in their technical report *Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description* [Broy *et al.*, 2009] : "The Functional Architecture describes the Total System from the black-boxperspective. It consists of a function hierarchy that contains the description of the functionality that is offered by the system to its outside world. Properties of the given hardware are not considered in this view."

USER INPUT The user can interact with the application and give different sorts of input. A component is dedicated to the graphical user interface and its connection with the haptic ergo-command:

- Free textual search: the user can give the name of an artist or a band in order to search for his/her/their album(s) and tracks
- List-based selection: results of textual search, the display of the tracks contained in an album are based on list that can be scrolled and selected, selection of subscribed podcasts.
- Information requests: The user can request the display of further information for an album or for an artist.
- Playlist-generation: When a track is being rendered the user can use it to request a playlist based on similar tracks (see 3.2.2).

Different screenshots of the personal radio prototype are detailed in appendix p. 219.

MEDIA RESOLUTION AND ACCESS When a media is selected for the playback, the client needs to decide whether the media is currently stored locally or need to be downloaded. Music media are based on a subscription to a charging service. We will not develop here on the different business models that are proposed on the market. Music providers use different DRM techniques, they have to assure the following functional requirements:

- only their customers with valid accounts can access content from their service
- only customers with valid accounts can playback content from their service (cf. media rendering).
- only authenticated device from the customer can store and playback the content
- content which is stored may not be transfered to other devices without permission

The media resolution functionality is capable of requesting a valid URL where to download the content and then to handle a download of the content. When different tracks are to be downloaded (album download, or playlist download), the player can start to playback the content. The tracks that have to be downloaded are queued and the media access component continues to download the following tracks.

MEDIA STORAGE AND INDEXING The media storage and indexing plays an important part in the architecture of this client. Actually, even if media are streamed directly from their sources, the personal radio always tries to download as much content as possible when a channel has been created.

MEDIA RENDERING The rendering component is responsible of the decoding of the audio content. It converts the binary information to an audio and video signal. Different codecs have to be supported since different providers are being taken in consideration. In this first prototype, the client does not offer the possibility to suspend/resume the rendering or to fast forward/backward since those functionalities would require a lot a development regarding the user interface, can be disruptive for the driver and have little interest in terms of architecture validation.

#### 4.3.1.2 *The technical architecture: software and hardware architectures*

The technical development of the prototype involved a lot of people and software coming from different sources. The goal was to create a software close to the one used in series vehicles. However for fast development purposes and to reduce costs, we designed both a vehicle environment to make demonstrations, and a desktop environment to test and develop the sofware before integrating it in a vehicle.

- Open-source software: OS for the business logic of the client (GNU Linux)
- Commercial software: OS for the simulation of the user interface (Windows), OS of the vehicle head-unit (QNIX)
- BMW series and pre-series software: user interface software as described in section 2.9.3.1
- Software developed for the project: business logic middleware
- Software developed to extend the project for research needs: business logic middleware

We use in the following the notion of technical architecture as defined by Broy *et al.* [2009]: "The Technical Architecture describes the Total System from the realization perspective. It describes, how the system that is specified by means of Logical Components can be integrated into a given hardware platform. It therefore consists of three parts: The Runtime Model, the Allocation, and the Hardware Topology."

We will here describe the different software and hardware components that were involved in order to illustrate the dependencies between the components that we have introduced in section 2.6.

THE USER INTERFACE The user interface is based on the Remote HMI (see section 80). The logic was developed as an application of the application server.

The Remote HMI does not support the transfer of audio and animated video data from the server to the client, so we had to use another interface to enable audio output in the vehicle. For the audio we used a direct jack connection to the head unit between the embedded PC that contains the business logic and the head unit which is capable of sending the audio signal to the loudspeakers. For the video we had to use another solution presented further.

The decoding of the audio media is based on the gStreamer[4] framework. Two codecs were used, the mp3 codec and the HE-AAC codec in order to be able to support content coming from different sources.

---

[4] http://gstreamer.freedesktop.org/

THE PROFILER Every user has a profile which is stored on the server. This profile contains the channels which have been subscribed by the user and the listening history of the user. The role of the profiler is to get the profile informations when they are synchronized with the middleware and to commit to the local storage all the changes in the profiles which are performed by the user in the vehicle.

THE MIDDLEWARE The middleware consists mainly in the HTTP interface between the vehicle and the remote services, *i.e.* the server backend for all the metadata services and the content providers for the direct media access. The middleware has two roles:

- exchange metadata information with the metadata backend: user profiles, subscribed channels, user search requests/results
- download content from the third-party content provider using the metadata delivered by metadata backend

The exchanges between the client and the server perform over SOAP since it is a standardised protocol, over a REST architecture where both the client and the server are stateless. It has been decided during the project to use a BMW proprietary XML dialect for the messages exchanged between the client and the server, and to encapsulate it in SOAP.

THE VEHICLE NATIVE DEMONSTRATOR The vehicle native demonstrator is using a CIC-High which is the cutting-edge head-unit of BMW series (see figures 26 and 27). The user interface runs completely on the head-unit which communicates over ethernet with a PC running a Linux distribution. The head-unit displays the infotainment application on the car infotainment display (CID)-display over an LVDS cable and handles events of the iDrive ergo-knob sent over a CAN bus. The Linux PC is connected to an UMTS router which is itself connected to the internet.

This client is a semi-embedded software and is close enough to the capabilities of an infotainment application in today's series production to be considered as a realistic demonstrator. However the Remote-HMI deployed on the CIC-High does not support software audio/video codecs which is a major drawback for an infotainment application

Therefore, the decoding component of the business logic is deployed on the Linux PC on which the gStreamer framework can be compiled. With this solution it is not possible yet to display video to the user. In order to solve this issue, we had to prefer to leave the head-unit platform for a complete simulation of the head-unit.

THE PC SIMULATION In order to be able to develop without testing on a vehicle head-unit, the remote-HMI can be runned in a simulation environment on Windows. Moreover, running the remote-HMI in Windows, permits the use a video client. Since the business logic still runs on Linux, the media files had to be exchanged between the two operating systems. For this, it has be decided to implement a UPnP accesspoint into the Linux business logic and then to develop a small client for Windows which pops up whenever video playout is needed. The usage of UPnP architectures has already been developed by Chekir [2007]. For this reason we will not develop much on this.

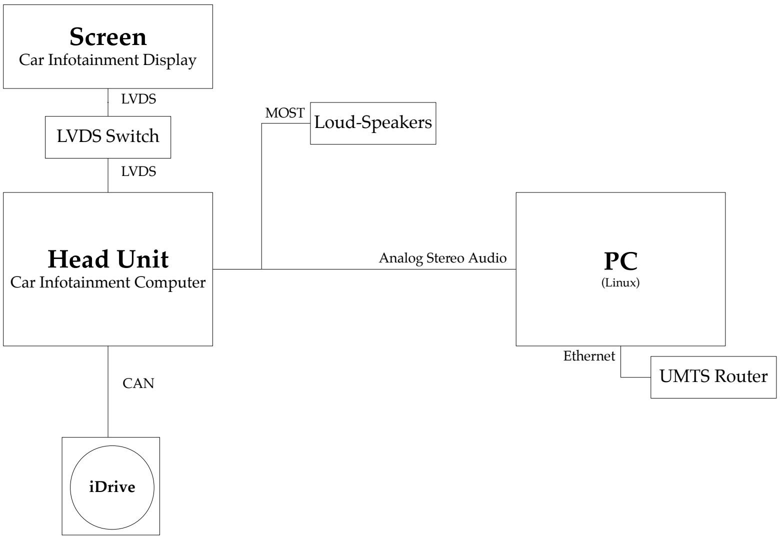


Figure 26: Client hardware architecture

Figure 27: Personal Radio in a vehicle

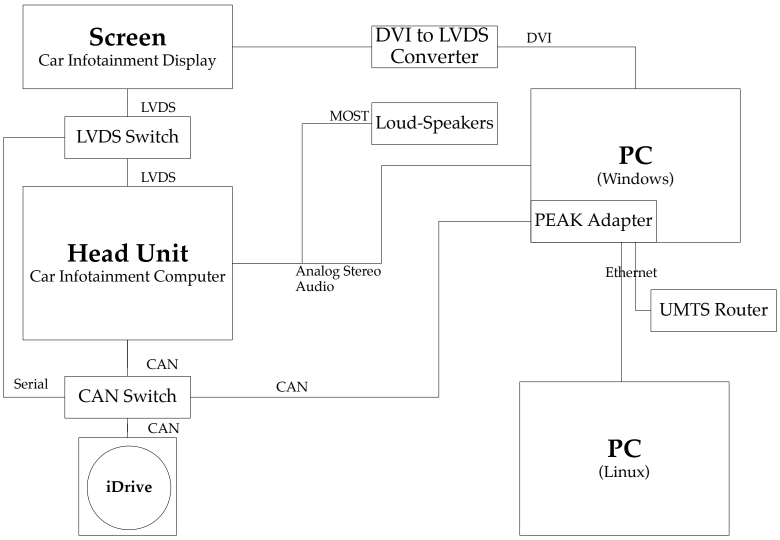


Figure 28: Client hardware architecture using HMI simulation

Since the Windows PC replaces the head-unit from the former architecture, it must have interfaces with the ergo-command and with the display. A PEAK adapter hase been used to be able to connect the Windows PC to the CAN bus and a DVI to LVDS converter was used for the video signal (see figure 28).

Regarding the mobile network connection, the technical architecture is the same as for the vehicle native demonstrator.

### 4.3.2 *The backend and frontend servers*

As we mentioned previously, we are speaking of actually two servers with distinct functionalities (see figure 29): (1) a backend server which handles the metadata and (2) a frontend server which handles different administration functionalities.

#### 4.3.2.1 *Service provider abstraction through the backend server*

As we mentioned in the section 2.4, there are many content providers for music, podcasts, videos, *etc.*. Technically, most of them have common basic functionalities such as text-search or content-download as we noticed when analysing Rhapsody but they have very different interfaces. This would involve a too big software complexity for the vehicle and would imply complex update mechanisms when a content provider changes its interface. For those reason the backend implements an abstraction mechanism: (1) A single interface between the backend and the client is developed; (2) The backend implements all the interfaces of the different providers. As a result, if an interface to a specific content provider need to be changed, only the client application is updated.

THIRD-PARTY METADATA SERVICE ABSTRACTION The channels and channels requests are expressed in an XML dialect (see Appendix p. 220). This dialect is rather close to MPEG-7 but does not rely on this standard for reasons only belonging to BMW internal development process. An XSD schemata has been designed (see Appendix for some exemples of the messages exchanged between the vehicle client and the backend).

- Music services: Specific adapters for each music on-demand service. The personal radio was tested agains two different content providers: Rhapsody[5] and Omnifone[6].
- Podcast services: A generic adapter capable of aggregating XML feeds from different sources[7]

INFOTAINMENT CHANNEL GENERATION In personal radio, channels can be three different types of infotainment playlists:

- music playlist containing only music
- information channels
- a mix of music and information

The infotainment channel generation is based on input given either by the user in the vehicule ('play more like this' functionality) or channel generation using the server frontend (see section 4.3.3).

PERSISTENCY OF THE USER CHANNELS The user channels are stored in the backend and synchronized when the client connects to the backend with the profile of the user.

### 4.3.3 *Server Frontend: User management and Channel Generator*

The server frontend is a web interface to be used with a browser. It is based on portlets running in the glassfish portlet container.

- *Administration portlets – User and Channel Administration Portlets.*Administration of channels (subscription to a podcast, deletion) and users can be added or removed.
- *Communication with the backend.* The frontend does not have any persistency mechanism. It communicates with the backend over the same SOAP interface as the vehicle client, in order to save administration operations in the user database.

### 4.3.4 *Application: creating a customer relationship management channel*

In order to illustrate the personalisation possibility in terms of media delivery, we have extended the former mention architecture with the following scenario: The customer relationship management of BMW wants to communicate directly in the vehicle about BMW products. Since each vehicle has an identifier and each customer of BMW connected services is registered, it is possible to personalise the advertisement. For instance, for people living in a winter snowing region, advertisement for winter tires can be sent to users. The language of the advertisement

---

[5] www.rhapsody.com

[6] www.omniphone.net

[7] Most of podcast service are using RSS 2.0 feeds, but some others use other XML dialects such as Atom.

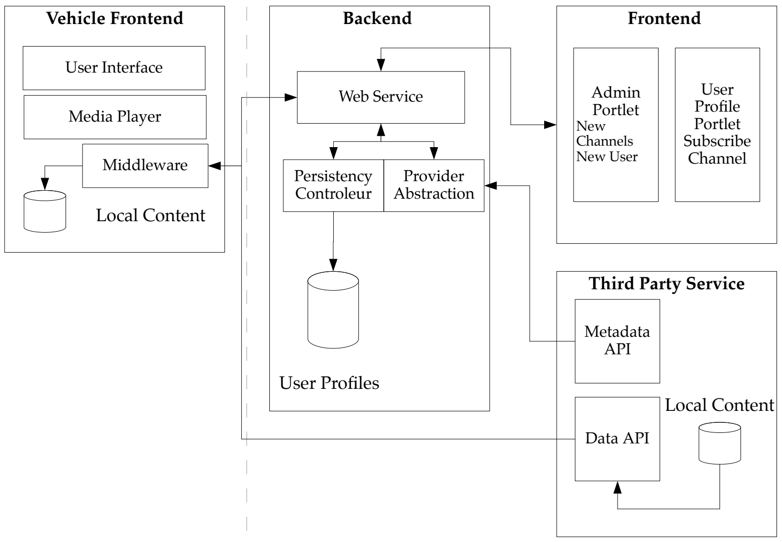


Figure 29: Functional architecture of personal radio

can also be personalised to the user. BMW had already a web based platform for the display of product information, in the vehicle using the vehicle browser (the BON browser). The purpose of our integration is to be able to redirect the users from the media consumption application to the product information platform (see figure 31). The technical architecture remaining the same as in the previous section, we had to extend the personal radio framework with the functional components:

#### 4.3.4.1 *Edition of personalised announcements*

We have developed an extension of the online platform for the creation of personalised product announcements. An announcement contains a title, a description of the product to be displayed, a text to be spoken and a clickable link for the BMW product information. Each announcement is given a set of filters which are used to determine if the annoucement is to be played within a channel or not.

#### 4.3.4.2 *Generating personalised channels*

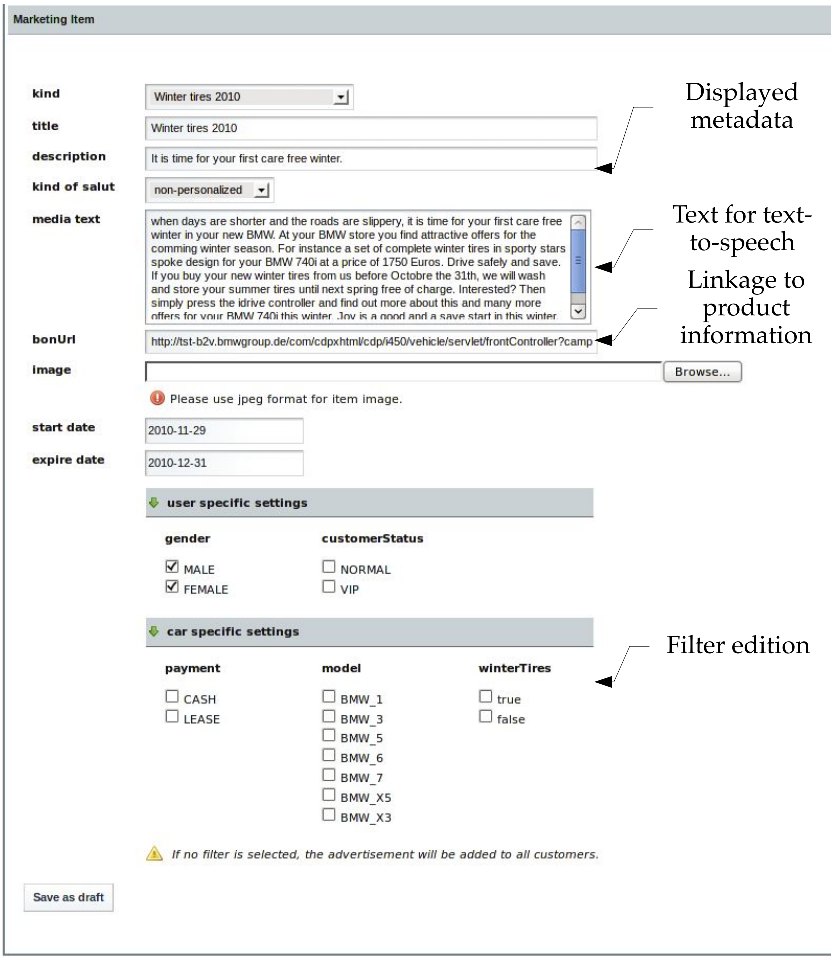
We first used text spoken by an actor for the advertisement but in order to make the creation process cheaper and easier we implemented a text-to-speech mechanism, so that advertisement from raw text could be directly generated. For this we used a text-to-speech interface from an external service provider. We have experimented three alternatives in order to test the genericity of the solution: The Mary TTS solution [8], the Google API[9] and the prototype of AT& T[10]. A big advantage of the TTS solution is that it is possible to create announcements directly addressed to the customer with his name which gives a more attentioned impression.

---

[8] We could integrate the API directly with our portlet: http://mary.dfki.de/

[9] At the time of our experiments, there was an unofficial API at http://translate.google.com/translate_tts which was capable of generating spoken content for small text sentences

[10] Unfortunately AT& T does not provide an official API for their service: http://www2.research.att.com/t̃tsweb/tts/



**Marketing Item**

**kind** Winter tires 2010 ▾

**title** Winter tires 2010

**description** It is time for your first care free winter.

**kind of salut** non-personalized ▾

**media text** when days are shorter and the roads are slippery, it is time for your first care free winter in your new BMW. At your BMW store you find attractive offers for the comming winter season. For instance a set of complete winter tires in sporty stars spoke design for your BMW 740i at a price of 1750 Euros. Drive safely and save. If you buy your new winter tires from us before Octobre the 31th, we will wash and store your summer tires until next spring free of charge. Interested? Then simply press the idrive controller and find out more about this and many more offers for your BMW 740i this winter. Joy is a good and a save start in this winter.

**bonUrl** http://tst-b2v.bmwgroup.de/com/cdpxhtml/cdp/i450/vehicle/servlet/frontController?camp

**image** Browse...

Please use jpeg format for item image.

**start date** 2010-11-29

**expire date** 2010-12-31

**user specific settings**

**gender**
☑ MALE
☑ FEMALE

**customerStatus**
☐ NORMAL
☐ VIP

**car specific settings**

**payment**
☐ CASH
☐ LEASE

**model**
☐ BMW_1
☐ BMW_3
☐ BMW_5
☐ BMW_6
☐ BMW_7
☐ BMW_X5
☐ BMW_X3

**winterTires**
☐ true
☐ false

*If no filter is selected, the advertisement will be added to all customers.*

Save as draft

Displayed metadata

Text for text-to-speech

Linkage to product information

Filter edition

Figure 30: Integration of customer relationship management in the personal radio

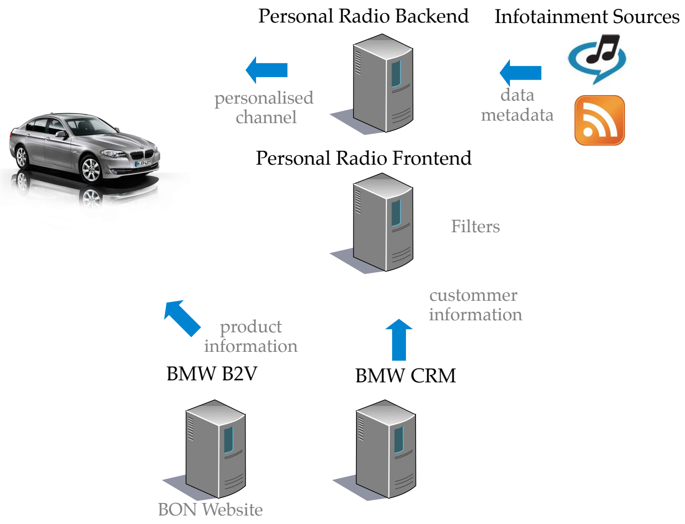


Figure 31: Integration of customer relationship management in the personal radio

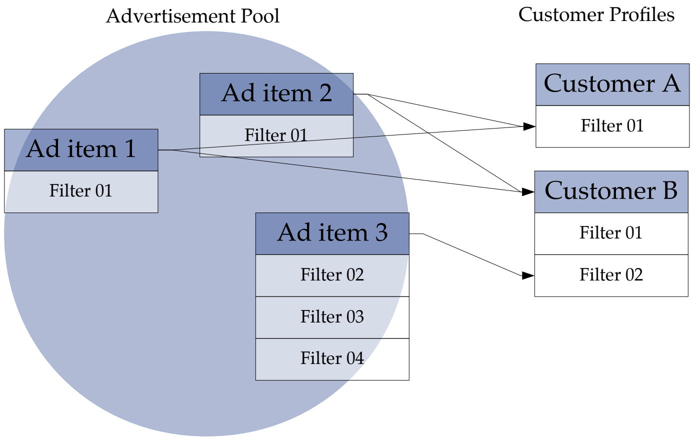


Figure 32: Advertisement selection

#### 4.3.4.3 *Personalised selection of advertisements*

When a user is playing an information channel, product advertisements are added to it at given time slots along the channel playlist. Those ads are personalised which means they meet the filters that have been selected (type of vehicle, age and sex of the driver) and they address directly the driver with his name. The selection process of ads is quite simple and is based on user description metadata (see figure 32 and see chapter 3). For each user, the ad characteristics are tested against the user description. Only the ad meeting all the filters within the user description are selected and added to the timeslots along the playlist. The selection process occurs remotely when the channel is generated. We will see in chapter 6 that we extended the decision process to the vehicle in order to use geolocalisation of specific ads.

## 4.4 CONCLUSION

Web based architecture are very convenient for the provision of content on-demand. They outperform broadcasting techniques and are still easier to use than peer to peer architectures. Unfortunately, web based services are not yet standardised and each content provider tend to implement its own public interface. When designing a rich client integrated with today's vehicle hardware and software as we have shown with the personal radio vehicle frontend, developing interface for every single content provider would generate too much software complexity and would raise updating issues. The abstraction of service through a backend is a solution which helps maintaining the vehicle software as generic as possible. In order to illustrate this flexibility of this architecture, we have extended it with a third sort of media content: personalised advertising. However, the solution that is presented in this chapter gives an answer to the media delivery problem. Regarding personal choice far from giving the vehicle driver the complete freedom of choice and does not exploit yet the whole potential of online media assets.

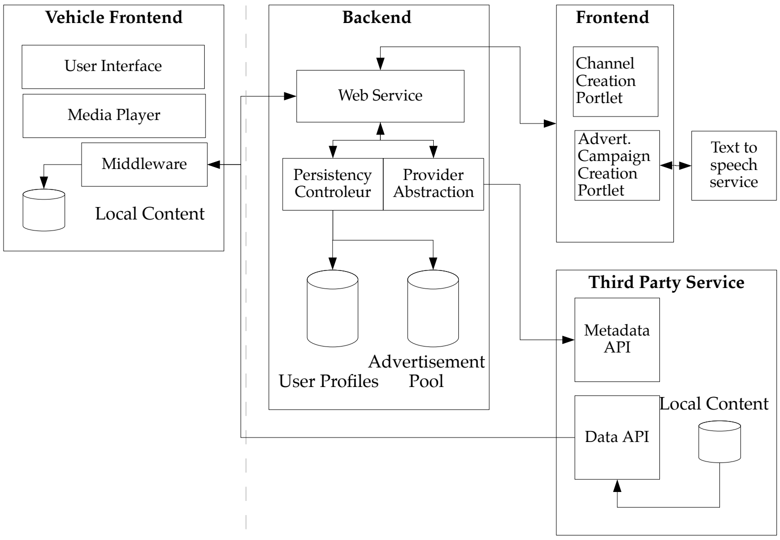


Figure 33: Extension of personal radio for the CRM scenario

5

# INTERACTIVE USER PERSONALISATION: BROWSING MUSIC RECOMMENDATIONS IN A VEHICLE

> *" Le but du jeu, c'est qu'à n'importe quel moment de la journée, l'auditeur lambda qui va se brancher, il ait son hit. Et ça, c'est impératif ! Si dans le quart d'heure, il n'a pas son hit, vous êtes mort ! Vous êtes mort ! Le mec, il n'a pas son hit au moment où il branche la radio, il va sur la concurrence."*
>
> — Philippe Mouricou, *Toujours la même chanson , propos d'un programmateur d'une radio musicale régionale française*[1]*, 2006*

After introducing the on-demand delivery of media content which presents a lot of potential in terms of personalisation, it is necessary to analyse how we can concretely personalise the infotainment experience. This chapters gives a solution to the personalisation issue when taken as an interactive problem between the driver and the vehicle media system. A special focus is given to the music which is the most consumed media in vehicles. This chapters analyses the ergonomics of a user active media search and retrieval. It explains the design of an online music browser prototype for a vehicle use, and its integration in the vehicle. We will analyse how the distribution of metadata aggregation, playlist generation and preview generation components can be adapted to the mobile delivery platform that we have defined.

## Contents

---

[1] "The rule of the game is that whenever he turns on the radio, the average listener gets his radio hit. This is a must! If it ever happens that he does not get his hit within the first fifteen minutes, you are finished! you are out! The guy who doesn't get his hit when he turns on the radio, what does he do? He switches to competitor broadcasters." Philippe Mouricou, *Toujours la même chanson*, excerpt of an interview with a radio playlist programmer at a french regional broadcaster.

Results of this chapter have been published in [Turlier *et al.*, 2010] and [Turlier *et al.*, 2008].

## 5.1 POSITION OF THE PROBLEM: "WHY MORE IS LESS?" IN A VEHICLE

Providing content on-demand liberates the user from the scheduled radio and TV programs which developed the so called 'channel hopping' behaviour among listeners who switch between broadcasters in search of the music they like. However according to theory Anderson [2008] *The long tail: Why the future of business is selling less of more*, the internet channel should enable its users to access products (respectively music tracks) which are not represented in mainstream outlets (respectively mainstream radio programs). Those tracks belong to the long tail. There is still a long way to go before true long-tail business models will appear. By analysing logs from LastFM[1] and MySpace[2], Celma proved that 0.28% of the total of artists account for 50% of the playcounts (number of tracks being played), and that 14% of them hold 86% of the playcount. On the one hand the music consumption is actually still driven by the 'hit' paradigm and on the other hand people always want to access a more diverse content.

As a matter of fact, this freedom to chose has a serious drawback as *Schwartz* demonstrated in his book *The paradox of choice: Why more is less* [Schwartz, 2005]. Now that the driver can select tracks among the millions of digitalised musical artworks and podcasts which are available online ; he might need some help in order to make his mind and decide what he could listen. We will focus in this chapter more specifically on music content, since listening to music is the preferred activity in the vehicles [Dibben and Williamson, 2007]

Indeed online music content has two characteristics which make it peculiarly difficult to use:

- There are very various types of music: which makes it a specially difficult to differentiate *a priori* between what one does like and what one does not like. As we have seen in chapter 3, there is no perfect taxonomy of them.
- The music content is updated very often with a high variance among the different sorts (some genres get new content every day, some every month): Therefore, it is necessary to take different types of users into account when designing a personalisation system; Somebody interested in the last music trends, will listen to a wide range of genres in order to satisfy his curiosity; whereas, someone focused on a rare type of music will expect to find all

---

[1] www.last.fm
[2] www.myspace.com/music

the music tracks available for a specific genre even if they are outdated.

We need to append to those issues the characteristics of mobile devices which are limited in terms of network and computing resources, and the specificity of vehicle interfaces that we have presented in section 2.9.2. The research on human computer interactions (HCI) for music applications has been in the last ten years a very active topic triggered by the constant improvements of music features extraction achieved by the audio signal analysis community. Lee [2010] pinpointed the lack of coordination between information retrieval research and application design research.

This chapter is about the design and the architectural integration with the modalities of a vehicle, of a new interface to create music playlists. We want more specifically to pursue the following goals:

- Design a graphical interface which is compatible with the ergo-knob and the display of a vehicle head-unit (see section 2.9.2)
- Aggregate information from online service providers, in order to achieve recommendation along different types of criteria
- Develop a scalable architecture in order to achieve functional requirements (multimodality, reactivity, unobtrusiveness)

## 5.2 BROWSING AND SELECTING MUSIC IN A VEHICLE OR MAKING THE USER ACTIVE WITHOUT DISTURBING HIM

Searching for music and creating playlist represents a potential workload for the driver which is bigger than just simply switching the radio on the head-unit. When designing a human interface for the vehicle it is very important to take into account the cognitive load of the tasks associated to its usage, since they have a strong influence on the security.

### 5.2.1 *Psychological background: Why and why not creating music playlists in a vehicle?*

When considering the psychological environment of a driver, we have to balance advantages and drawbacks of the use of infotainment systems in a vehicle.

#### 5.2.1.1 *Listening to appropriate music is entertaining ...*

Listening to music in a vehicle can have a positive effect on the driving performance. Music can be either stimulating and help to fight against boredom or relaxing when traffic conditions are getting bad by making the driver. Sloboda *et al.* [2001] have established that music makes listeners "more positive, more alert, and more focused". Dibben and Williamson [2007] even speak of a 'self-therapy': "listening to music while driving can be seen as means for individuals to alter their environment in a way that is appropriate to their needs".

#### 5.2.1.2 *... but browsing music can be obtrusive*

However, infotainment system if they are not properly use can negatively influence the driver capability to drive. Lansdown [2000] pointed

Figure 34: Spatial distribution of user tasks in a vehicle (image BMW)

out that the integration of the in-vehicle infotainment (IVI) systems is a major source of user distraction for the following reasons: the loading time of internet applications can be "long and unpredictable" and internet information is "dynamic and inconsistent". When driving, a person has multiple parallel tasks to manage which can be classified in three classes according to Geiser [1985]. They are distributed over three zones within the interior space [Tonnis *et al.*, 2006, Gebhardt, 2010] (see figure 34).

1. Primary Tasks (directly vehicle steering related tasks): *e.g.* acceleration/deceleration, watching the driving environment and at other traffic participants (vehicle, pedestrians, *etc.*).
2. Secondary Tasks (indirectly vehicle steering related tasks): *e.g.* operating the windshield wiper or setting turning signals.
3. Tertiary Tasks (comfort tasks not relevant for the steering of the vehicle): *e.g.* gps navigation, radio tuning, air conditioning, *etc.*

Infotainment-related tasks are obviously belonging to the third category

DRIVER DISTRACTION Not only are the shortcomings of IVI systems sources of obtrusivity by themselves (and thereby of insecurity) but moreover they can lead to user frustration and stress which is a second source of insecurity [Matthews *et al.*, 1996]. Burns insists on the visual distraction which is imposed by those systems and emphasizes on the fact that a proper structure of information is necessary [Lansdown, 2000]. There are four sorts of driver's distraction [Dibben and Williamson, 2007] :

1. The visual distraction is caused by the visual load necessary when looking at the infotainment application in the vehicle;
2. it is coupled with the biomechanical distraction caused by the manipulation of the equipment which can prevent the driver to change gear correctly.

3. The auditory distraction is caused by the audio masking of vehicle engine noise, street noises and warning signals that can prevent the driver to take information on his environment.
4. The cognitive distraction is due to the reflexion necessary when using a complex infotainment system (*e.g.* find options, search content).

### 5.2.2 *Making user interactive media interfaces more usable and less frustrating*

The good practices in terms of vehicle interactive applications design have been exhaustively listed by Lansdown [2000] who made a meta-study of human factor solutions. In the design of our music browser solution, we will focus on the following aspects identified by Lansdown [2000].

- "any navigation-system task [...] should not take more than 15 seconds to complete when measured as a continuous task"[Green, 1999]
- the average glance duration should be less than 1.2 seconds [Dingus *et al.*, 1989]
- No glances should be longer than two seconds [Zwahlen *et al.*, 1988, Zwahlen and DeBald, 1986]

Giving a proper structure to information in a vehicle means optimising the use of the different modalities in order to minimise the cognitive load of the driver [Sarter, 2006]. There are only three modalities that can be used in order to interact with a rich media system in a vehicle: visual, audio and haptic. The following subsections are summarised in the table 9.

#### 5.2.2.1 *Visual modalities*

It is necessary to use additional information to browse the music catalogues ([Chen and Butz, 2009]). The visual modality is certainly the richest in terms of information that can be transmitted to the user.

TEXTUAL INFORMATION Textual information can be used as input (formulation of the query) as well as output (presentation of the result). As we have seen in chapter 3, most of the metadata for music are textual data. Commercial music browsers use free-text search querying system where the user can enter any query string he likes or predefined categories (like for instance Spotify in figure 35).

Textual information is efficient to describe precise concepts without ambiguity. However, Kim and Belkin [2002] have shown that the conceptual categories have a very unequal importance among the different categories of classification (see table 8). Information like emotions or context of use turn out to be the most expressive textual information that can be used to build a music search engine based on semantic tags.

In order to help the driver to find out relevant information in a text, several techniques based on the size of the text, its orientations, the color and the contrast can be used [Lansdown, 2000]. Desktop applications for music use mainly lists (like Rhapsody in figure 36).

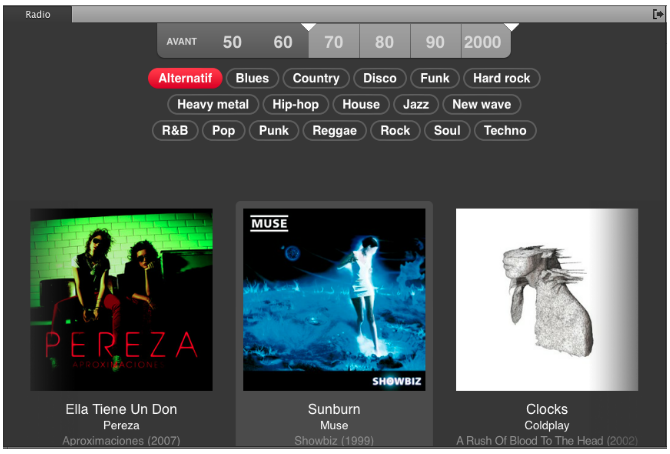
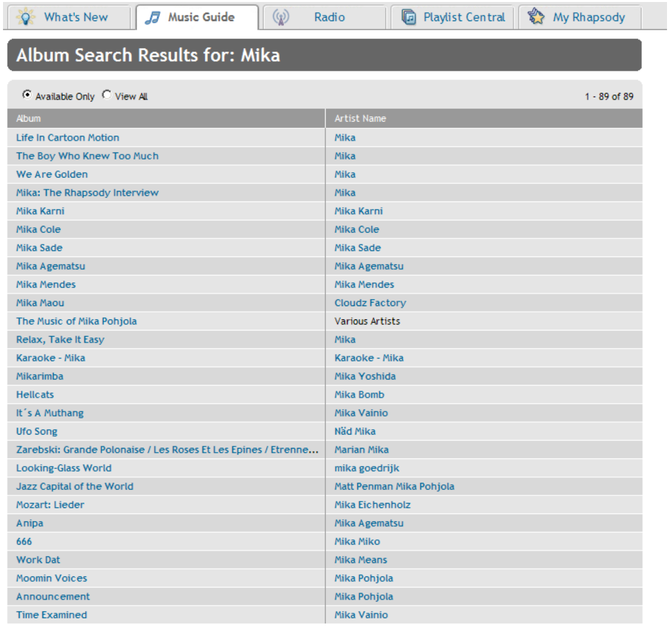


Figure 35: Spotify music genre and time period selector

| Categories | Explanation | Freq. | Examples |
|---|---|---|---|
| Movements | Words related to specific movements | 3% | Running away; Flying; Sprint |
| Neutral concepts | Words that are evaluatively ambiguous or neutral | 19% | Ambivalence; Transformation; Simplicity; Realization |
| Emotions | Words explicitly indicating emotional status | 24% | Happy; Joyful; Sad; Threat; Cheerful |
| Nature | Words indicating nature-related phenomena | 10% | Nature; Trees; Flowers blooming; Bees; Butterflies |
| Objects | Words indicating concrete materials other than nature | 2% | Spy; Europe; Wizard; Queen Elizabeth |
| Occasions or filmed events | Words describing specific or events – also referring to filmed events | 29% | For celebration; For Baroque party; Grand arrival or entry; Song for exploring forest; Saturday at the Art gallery |
| Musical features | Words indicating musical features | 13% | Violin; Slow-tempo; Orchestra; Rondo; Strings; Symphony |

Table 8: Importance of textual semantic categories for the music searching task (study of Kim and Belkin [2002])



Album Search Results for: Mika

● Available Only ○ View All 1 - 89 of 89

| Album | Artist Name |
|---|---|
| Life In Cartoon Motion | Mika |
| The Boy Who Knew Too Much | Mika |
| We Are Golden | Mika |
| Mika: The Rhapsody Interview | Mika |
| Mika Karni | Mika Karni |
| Mika Cole | Mika Cole |
| Mika Sade | Mika Sade |
| Mika Agematsu | Mika Agematsu |
| Mika Mendes | Mika Mendes |
| Mika Maou | Cloudz Factory |
| The Music of Mika Pohjola | Various Artists |
| Relax, Take It Easy | Mika |
| Karaoke - Mika | Karaoke - Mika |
| Mikarimba | Mika Yoshida |
| Hellcats | Mika Bomb |
| It´s A Muthang | Mika Vainio |
| Ufo Song | Näd Mika |
| Zarebski: Grande Polonaise / Les Roses Et Les Epines / Etrenne... | Marian Mika |
| Looking-Glass World | mika goedrijk |
| Jazz Capital of the World | Matt Penman Mika Pohjola |
| Mozart: Lieder | Mika Eichenholz |
| Anipa | Mika Agematsu |
| 666 | Mika Miko |
| Work Dat | Mika Means |
| Moomin Voices | Mika Pohjola |
| Announcement | Mika Pohjola |
| Time Examined | Mika Vainio |

Figure 36: List of results in Rhapsody

Figure 37: The music rainbow [Pampalk and Goto, 2006]

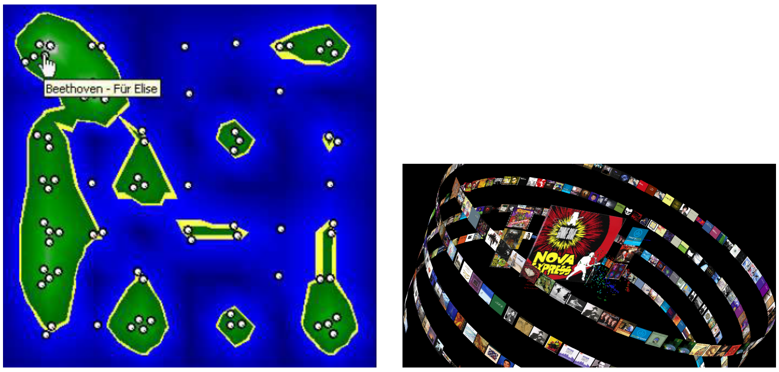
In a vehicle, text information is mainly based on lists which are very easy to manage using an ergo-knob (see section 2.9.2). Bederson suggests to use fish eye menus [Bederson, 2000] introduced by Furnas [Furnas, 1999] when displaying menu-lists on small screens to help the user to focus rapidly. An innovative solution has been proposed by the Music Rainbow [Pampalk and Goto, 2006] which is a prototype using a knob to select through aggregated music tags from social services (see figure37).

IMAGE INFORMATION Even if images do not have the precision of text, they have a much stronger emotional impact. [Vignoli, 2004] has shown that the visualisation of the album is an important information for the music choice. Desktop applications, as well as handled devices like iPod, make use of album covers, in order to increase the expressiveness of their user interface.

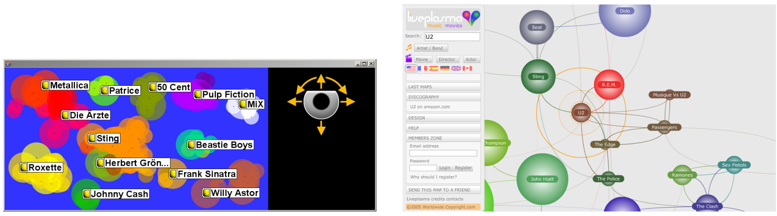
OUTPUT/INPUT: TOPOLOGICAL INFORMATION A large number of research projects have tried to display music catalogues on 2D or 3D spaces in order to give a topological structure of the content. Similar music tracks are grouped together or presented as part of a network like in live plasma (see figure 38d). One of the first systematic attempts has been achieved by Pampalk who focused on signal feature extraction and clustering techniques in order to design 'islands of music' (see figure 38a). Pampalk suggests to extract features from the signal in order to build a self-organizing map. A similar approach based on signal analysis and dimension reduction using singular value decomposition has been taken by [Chen and Butz, 2009] who combined it with region selection techniques so that the user can select several tracks on a map. Lamere and Eck presented a 3D visualisation technique (see figure 38b).

In the automotive domain, Rölle suggested another approach based on a predefined taxonomy (a genre tree): The Music-Map (see figure 38c) can display the music data stored in an iPod using the ID3 tags of the music. Rölle adapted the use of the iDrive knob in cartographic applications (GPS navigation) to the navigation in a local catalogue of an iPod.

This approaches have the following advantages: (1) they fast give an overview of the 'big picture' to the user who can clearly identify 'masses' (which parts of the catalog are well represented and which are not), (2) they provide a 'navigation' paradigm similar to the map navigation which is familiar to a lot of drivers. However they have some major drawbacks: All the solutions which have been suggested work only with local data. As a matter of fact, they are all based on the calculation of clusters or classes, and thus require offline algorithm to be processed. A solution can be to use a predefined taxonomy like in Music Map but it reduces also considerably the dimensions of personalisation and the adaptation to change in the online catalogue.



(a) Islands of music [Pampalk, 2001]
(b) 3D visualisation [Lamere and Eck, 2007]



(c) BMW Music-Map [Rölle, 2007]
(d) Liveplasma

Figure 38: Topological representation of music catalogues, images reproduced with authorisation

Semantically orthogonal concepts, can be presented in multi-dimensional spaces where the user can position his query using a cursor. For instance emotions (music mood see section 3.2.2.1) can be represented on a 2D space like the energy-stress Model of Thayer [1989] and the valence-arousal model of Russell [1980]. Such input is used in the

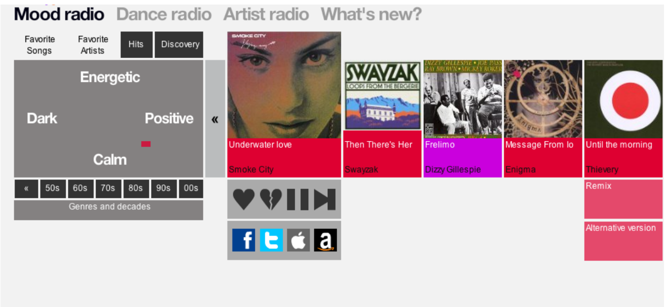


Figure 39: 2D input for semantically orthogonal concepts in musicovery

Musicovery (see figure 39) where the user can specify valence (degree of attractiveness or aversiveness) and the degree of arousal (intensity) .

### 5.2.2.2 *Audio modalities*

When speaking of music, the audio modality plays a very important role. Music is decoded and rendered through the speakers the vehicle. We have seen in section 2.9 that the adaptation of the volume to the speed of the vehicle is already integrated in nowadays vehicles, in order to avoid audio masking. However, the use of the audio has a modality in vehicle infotainment systems goes far beyond the playback of music since it can be use as input as well as output.

**INTERFACE FEEDBACK : AUDIO MENU CUES** Sound cues can be used to deal with visual information overload in media systems. Jeon *et al.* [2009] made a user study comparing different types of sound information (and their combinations) that can be used in vehicle list menus:

- Compresses speech sounds (or spearcons): They are short sounds, generated by timely compressing spoken phrases
- Speech indexes (or spindex): They are produced by giving each menu item a auditory cue based on the pronunciation of the first letter of it.
- Time to speech (or TTS): They are computer generated pronunciation of the menu items

The user study of Jeon *et al.* [2009] is based on a dual task (a primary task being a simple game and the secondary being finding a track in a list of 150) which is unfortunately not vehicle specific. Nevertheless, they demonstrated a noticeable improvement of the primary task (success rate of the game increase by more of five points) when using TTS, TTS and spearcons, TTS and spindex or even the three all together (TTS and spearcons and spindex); moreover they noticed a diminution of the time to find music tracks when using them. The study of Jeon *et al.* [2009] gives a very interesting solution for the browsing of long playlists in a vehicle. However, it does not prove that it could have a significant contribution for shorter playlists and even if they are still valid with longer lists.

**INPUT: SPEECH RECOGNITION** Speech recognition has been integrated very early in mobile phones. Speech recognition as a modality of input has been a much debated topic in the vehicle HMI community. On the one hand, using the speech as input avoids overloading the haptic modality but on the other hand speaking in a vehicle is generally considered as a very strong cognitive limitation, since it generates resource competition [Lee *et al.*, 2001, Strayer *et al.*, 2003].

INPUT: QUERY-BY-HUMMING In-stead of using speech recognition, melody recognition also known as query-by-humming has been suggested as we had seen in and some research prototypes already exist[3] . The recognition is based on signal processing (pitch extraction and rhythm extraction). This technique is very useful to find a very specific song when the name of the artist and of the song are unknown to the user. It is believed that human beings can remember melodies much better than names and than lyrics. Regarding the vehicle ergonomic, it turns out that query by humming requires long 'humming' to work properly which would generate to much cognitive load from the driver. Moreover the processing is very resource demanding and could not be implemented in a vehicle which would result in high network-load and poor reactivity.

#### 5.2.2.3 *Haptic modalities*

TEXT INPUT AND ITEM SELECTION Car manufacturers provide different techniques to input text and select item by using driver's hands either with buttons, knob or touch screen. Haptic user input is necessary when formulating the query (search a song, or searching a genre), selecting, results and starting playback. Depending on the type of search (free-text search) or categorical search, the cognitive workload can be different as we will further see in the description of the user tasks. In mobile applications as well as an increasing number of desktop applications, the use of predictive text input has spread since the introduction of T9 (technique using the nine keys of a cellphone) and new techniques are still being proposed [MacKenzie, 2002, Oniszczak and MacKenzie, 2004].

OTHER TYPES OF INPUTS Geiger [2003] studied in his doctoral dissertation a gesture-recognition technique for the use of head-unit's infotainment functionalities. This gesture recognition technique, uses infrared sensors (which measures in a 3D space the position of the driver's hand and of the driver's head). The results of Geiger are very promising for the navigation through lists (especially when using the so-called 'continuous gesture'). However, Geiger admits, that the continuous gesture technique requires a continuous visual contact with the display for the feedback, which results in a high visual load.

Music queries based on the rhythm recognition have already been implemented and are available on web services[4]. The user just has to tap the rhythm of a song. As for query-by-humming the recognition is based on the signal analysis of the different music tracks and has the same drawback regarding the long workload of the input. The recognition rate turns out to be quite low.

---

[3] An online web exemple is available at Midomi: http://www.midomi.com/, or for smartphones at http://www.soundhound.com/

[4] www.songtapper.com

| | Modality | In. | Out. | Type of Information | Advantages | Drawbacks |
|---|---|---|---|---|---|---|
| VISUAL | Text | | × | classification taxonomy (genre, mood) | precise, easy to browse (menu lists, gliders, *etc.* | not always illustrative |
| | Image | | × | illustration (cover art), artist portrait | emotional and illustrative | imprecise |
| | Topology | | × | masses (clusters of similar objects) | 'big picture', quantitative information | difficult to browse, offline algorithm |
| | Orth. Sem. | × | | cursor position | precise visual tuning | poor feedback, difficult to control |
| AUDIO | Speech r. | × | | text input, command | no haptic and visual load | disturbing (cognitive load for the formulation of the query) |
| | QbH | × | | melody | no haptic and visual load | long input and processing (bad reactivity) |
| | Men. cues | | × | metadata, music summary | not visual load | needs processing |
| HAPTIC | Textin. | × | | Text description | very precise | long cognitive load |
| | Item sel. | × | | Text or image selection | short cognitive load | less precise than text input |
| | Gest. | × | | hand and head gesture | min hapticload | need visual feedback |
| | Tapp. | × | | rythm of the music | minimal cognitiveload | imprecise, unreactive |

Table 9: Using visual, audio and haptic modalities to displays and select information

Figure 40: Gesture control of infotainment menus [Geiger, 2003]

#### 5.2.3 *Active personalisation of the consumption process: The user tasks*

In the consumption process that we have described formerly (see *infra* section2.3, we can identify two tasks which enable active personalisation of the user.

##### 5.2.3.1 *Find content*

The formulation of the query is a tradeoff between the time needed for the formulation and its granularity. The more specific is the filter, the better are the chances that the filter will achieve a relevant output. However, the more refine is the query, the bigger are the visual demand and the cognitive workload. Graf *et al.* [2008] have established that, both free text search and categorical search are suitable for a vehicle use. Free text search turns out to be faster when looking for specific items. The advantage of categorical search over free search is that the user does not need to know *a priori* what he is looking for, since the categories are themselves suggestions of input. Hence free text search in mobile devices, is more likely to be applicable to a locally stored content (which corresponds to the conditions of the study of Graf *et al.* [2008]).

The semantic search enables the combination of different categories. For music categories like genre, mood, hotness, time-period can be used to create filters. For information editorial vocabularies have been proposed to select topics, iTunes proposes for instance 14 categories and 50 topic subcategories for podcasts.

The selection of the result of the query is another aspect of active personalisation. The user can select among the different filtering results what are those who best match his idea. This part is of course very dependent of the way the music is displayed. In our former prototyped study the information items are displayed in list formats because it is a format where textual information can be maximised, for the music prototype we explored both textual lists and visual information.

The last aspect of the user active personalisation is the modification of the result. This can be done with the proposition of alternative recommendations which do not perfectly match the first query but which are linked to the result and thus can be selected by the user with a minimal workload overhead.

| Consumption Step | User Task | Modality | | | Driver distraction potential | Driver frustration potential |
|---|---|---|---|---|---|---|
| | | V. | A. | H. | | |
| **Discovery** | Watch query possibility | × | (×) | | loading time | choice size structure |
| **Query Formulation** | formulate query | × | × | × | input size | input structure |
| **Filtering**[5] | no tasks | | | | processing time | result relevance |
| **Preview** | Watch preview | × | | | information structure | illustration of preview |
| **Results selection** | Select result | × | × | × | interaction structure | interaction clearness |
| **Consumption** | Listen | | × | | reactivity | sound level |
| **Results modification** | Select and Modify | × | | × | interaction structure | reactivity |

Table 10: Distraction and frustration potential of a music browsing interface

#### 5.2.3.2 *Create playlist*

The creation of the playlist is the ultimate task that the user has to fulfil. We can see it as a specific application of the task consisting of finding content. The manual creation of a playlist in a vehicle by the repetitive process of find and select is not applicable at all. With the version of personal radio that we have presented in the former chapter, the creation of the playlist is based on a single track which has the advantage to minimize the user input. More refined playlist creation techniques require the combination of different criteria as input as we will see in the following sections.

### 5.2.4 *Analysis of the functional requirements of a vehicle music browser*

We can now analyse the potential distraction and frustration that can be brought by the interface in the different steps of the vehicle media consumption that we have presented in section 2.3 (*p*.67). Those different risks (see table 10) have to be taken into account in all the design steps of the solution: graphical design of the interface, architectural design of the application (logical, software and hardware architecture), integration in the vehicle.

The on-demand media consumption entails a great potential in terms of user active personalisation. Every step of the interactive process bestows the opportunity to the user to formulate his music preferences, discover music suggestions and select music tracks in order to make a playlist. However, as an interactive application, the music browsing is a sensitive application in terms of safety, this is the reason why we mapped visual, audio and haptic modalities to the different tasks, and

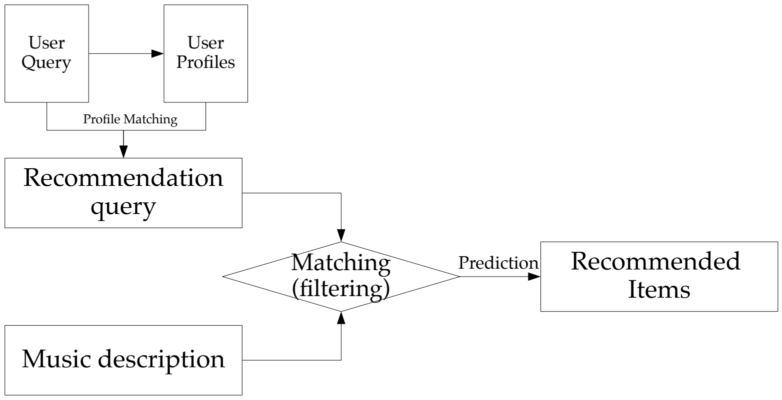


Figure 41: The recommendation problem (adapted from [Celma, 2008, p. 25])

analysed the different sorts of user workloads which are associated to them.

## 5.3 THE INTEGRATION OF THE "RECOMMENDATION" MAGIC IN A MOBILE SYSTEM

We have seen in the previous section that there is a hidden part in the infotainment consumption process: the filtering. Actually we are speaking of much more than filtering, since we aim at supporting the driver in his music choice when browsing the catalogue for the creation of a playlist: This is a recommendation problem (see figure 41). Two aspects are very relevant for the integration in a vehicle:

1. the quality of the recommendation (bad recommendation can be frustrating)
2. the processing of the recommendation (long processing is disturbing)

We will not develop the qualitative aspect of music recommendation since exhaustive studies have already been made (summarized by [Celma, 2008]) and this would go far beyond the scope of this thesis. We will rather focus on the processing of recommendation in a mobile architecture. Depending on the recommendation technique this processing may need more or less data input and computing resources. Moreover, since we are speaking of mobile devices we need to add the network resource as a variable in the design of our system.

### 5.3.1 *Architecture of recommendation browsing solutions: related works*

Before we present our solution to the vehicle browsing of musical recommendation, we can consider solutions of the literature. None of them is vehicle specific but they all at least share one of the following requirements that we target: (1) connectivity to internet (2) music specific browsing interface (3) integration of recommendation technique.

(a) Musicbox projection, selection of tracks along a path

(b) selected tracks

Figure 42: Track selection based on area selection in MusicBox [Lillie, 2008], images reproduced with authorisation

### 5.3.1.1 *Local music browser connected to the internet*

The Musicbox has been developed by Lillie [2008] at the MIT Media Labs. It consists of a desktop software composed of a windowed graphical interface, an agent capable of analysing a local music repository and two software interfaces with online metadata providers. When analysing a music collection, the Musicbox takes a wide range of values into account: (1) audio signal values; (2) expert classification and (3) social tagging. Audio signal values are either locally extracted or retrieved using a remote service (the EchoNest[6]) and the tagging comes from a social network (Last.fm[7]). Each track is represented by a vector, and a principal component analysis is performed on the dataset in order to compute a 2D projection (see *figure* 42).

Even if it only deals with local content of 1500 to 2000 songs, the architecture of the MusicBox (see figure 43) illustrates the aggregation and the combination of different sources of metadata from online services. Those services are directly integrated in the architecture, which means there is a tight relationship between them and the application.

No doubt that the Musicbox is a significant illustration of the synergy between multimedia and HCI research as described by [Lee, 2010]. However it still has strong limitations regarding a potential integration in a vehicle. Firstly, the local analysis or the uploading of local raw data is not thinkable for technical reasons (computational resources, and network resources) and music licensing reasons (no unauthorized third party may have access to licensed music data). Secondly, the user interface requires intentionally a lot of visual feedback : "Dynamic queries apply the principles of direct manipulation with a visual presentation of query and results; rapid, incremental control of the query; selection by pointing instead of typing; along with immediate and continuous feedback" which is not adapted to a vehicle.

---

[6] www.echonest.com

[7] www.last.fm

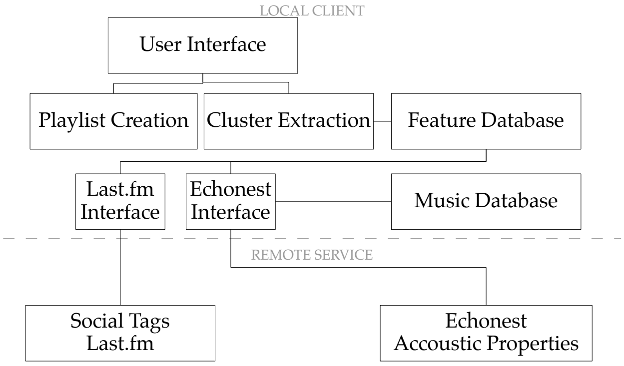


Figure 43: Functional architecture of the Musicbox

| | Musicbox | Vehicle integration potential | |
|---|---|---|---|
| | | Usability | Architecture |
| Discovery | Image colored density clusters | ☆★★ | ☆☆☆ |
| Query Formulation | Area or path selection | ☆☆☆ | ☆☆☆ |
| Filtering | Local calculation of PCM of acoustic, social features | | ☆★★ |
| Preview | No preview (local) | ☆★★ | ☆☆☆ |
| Selection | Track List | ☆☆★ | ☆☆☆ |
| Modification | filtering feature check in-/out | ☆★★ | ☆☆☆ |

Table 11: Vehicle integration feasibility evaluation of Musicbox browsing concepts

### 5.3.1.2 *Online music recommenders*

We have already mentioned some online services which are available through a web interface and deliver music recommendation. Most of them use a "seed song" or a "seed artist" to generate a playlist based on different similarity techniques like Pandora[8] or Last.fm [9]. They are very simple to use for someone who has a music track in mind and want to create a playlist which sounds like it (see "play more like this" in chapter4). The DBREC is research prototype has been realised by Passant and Decker [2010], Passant [2010] and is only based on linked dataset in order to explain how it recommends the music. However, all those examples request from the user to manually input the name of a track or of an artist and give few possibilities to the user to tweak the query when the result is not satisfying. The MusicLens[10] has another approach, it leaves to the user the tuning of the different parameters which have to be used. The tuning is based on vertical sliders as depicted in figure 44, which give to the user the possibility to formulate a query according to ten dimensions: volume (from silent to ear-busting), tempo (from slow to fast), voice (from vocal to instrumental), size (from vocal to orchestra), purpose (dance, sex, driving, listening), sex (female, both, male), age, mood (smile, angry), colour (rainbow rule), time (from very old to now). The query is presented to the user as a 3D image where each of the dimension influence the convexity of areas of a surface, the image is generated on the client user interface in order to give a fast visual feedback to the user. When changing any of the slider cursors, a new playlist is almost instantaneously generated and displayed as a list below the query. The MusicLens gives interesting architectural and interaction solutions to the interface problem for the playlist creation scenario. It provides a fine granular tuning interface, where the user can in a few incremental steps define a playlist query, watch the result and, if necessary, modify the original query in order to achieve the expected result. The query preview is generated locally whereas the playlist satisfying the criteria is computed remotely so that the client just has to display the result which saves processing and network resources. The rulers are easy to understand and adapted to the knob that is being used in vehicles. However, the interface of MusicLens does not make use of the full potential of music metadata. No image information is displayed in order to illustrate the different tuning categories, and the query viewer is difficult to interpret in terms of musicality. The granularity of the tuning categories is not displayed and gives to the user that it is the same among them even if it is not the case. Moreover, the result consist of simple list and does not suggest any alternative.

### 5.3.1.3 *Integration of music browsers in mobile devices*

Research about the integration of music browsers in mobile devices has tried to tackle the limitation of displays by various visualisation techniques: Goussevskaia *et al.* [2008] used a dimension reduction technique similar to the Musicbox. However, when to much computational effort is required, it seems necessary to transfer a part of the processing to an remote entity. This is actually the approach based on ontologically

---

[8] www.pandora.com

[9] www.last.fm

[10] www.musiclens.com

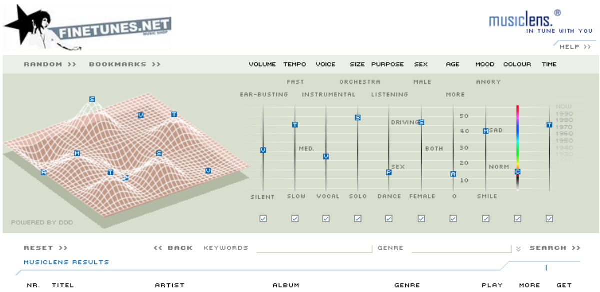


Figure 44: The MusicLens user interface

| | MusicLens | Vehicle integration potential | |
|---|---|---|---|
| | | Usability | Architecture |
| Discovery | Categorie rulers | ☆★★ | ★★★ |
| Query Formulation | Categorie tuning | ★★★ | ★★★ |
| Filtering | Remote high-level accoustic and semantic data | | ☆★★ |
| Preview | 3D plane for query | ☆☆★ | ★★★ |
| Selection | Track List | ☆☆★ | ★★★ |
| Modification | Categorie tuning check in/out | ☆★★ | ★★★ |

Table 12: Vehicle integration feasability evaluation of MusicLens browsing concepts

structured expert metacontent which has been taken by Noppens *et al.* [2007a,b] in their MobiOnt project when developing the MobiXpl radio browser. Noppens *et al.* [2007b,a]designed a radio discovery service, based on semantic definitions of radio editorial content. The graph structure of the ontology describing is translated to a SVG tree on a server adapter. The SVG representation can be used in a light-weight mobile phone application. The user can walk through the structure and select semantic definitions (see figure 46-a). When the user interacts with the client, instructions are sent to the adapter which controls the business logic (see figure 45). The user can specify direct preferences (“I like genre A”) or even complex preferences (“I like genre B more than genre A”) (see figure 46-b). When a filter does not give the expected result, a preference relaxation strategy is processed. It consists in proposing more general concepts (ontology parent concepts) in order to help the user in his query formulation. The interface of MobiXpl is using a topological structure (given by the graph of the ontology) and text elements information elements for semantic genres. It looks very relevant for a mobile phone, since the user can afford long lasting visual feedback when looking for a radio station, however it is not quite clear how such structures could be walked-through efficiently using the iDrive knob. Moreover, the visual modality is not fully addressed since no visual image is used in order to illustrate the categories.

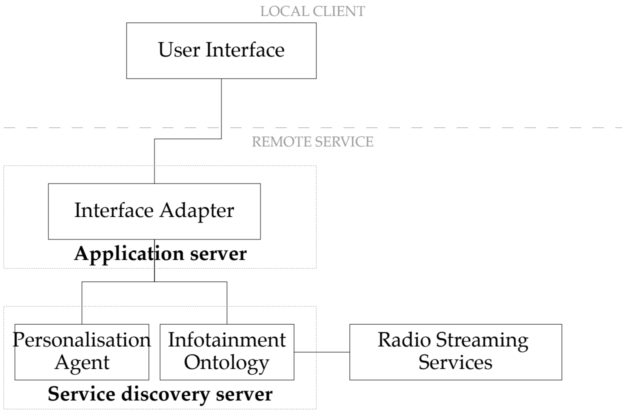


Figure 45: The MobiXpl layered architecture (adapted from [Noppens *et al.*, 2007b])

(a) MobiXpl: Gradual expansion of music semantic genres

(b) MobiXpl: Preference between Jazz and HipHop

Figure 46: MobiXpl on a Nokia telephone running Symbian 40 [Noppens *et al.*, 2007a]

| | MobiXpl | Vehicle integration potential | |
|---|---|---|---|
| | | Usability | Architecture |
| Discovery | Ontology (text nodes, edges) of expert data | ☆★★ | ★★★ |
| Query Formulation | Category selection | ☆☆★ | ★★★ |
| Filtering | Semantic graph | | ★★★ |
| Preview | No preview | ★★★ | ★★★ |
| Selection | Automatic | ☆☆☆ | ☆☆☆ |
| Modification | Preference relaxation | ☆★★ | ★★★ |

Table 13: Vehicle integration feasability evaluation of MobiXpl browsing concepts

### 5.3.2 *Heuristic evaluation of the concepts*

We have specified in tables 11, 12 and 13 a heuristic evaluation of the aforementioned solutions. There is no single model for user recommenders especially in the music domain as Swearing and Sinha [2002] demonstrated in a user study. They stated that "an effective recommender system inspires trust in the system; has system logic that is at least somewhat transparent; points users towards new, not-yet-experienced items; provides details about recommended items, including pictures and community ratings; and finally, provides ways to refine recommendation by including or excluding particular genres". Some recommenders give extensive information about the results they yield such as the DBREC but their user interface is far too desktop specific for a vehicle use. Giving the user the possibility to tweak the results as with the MusicLens is definitely a key functionality which has strong consequences on the interface (multiple criteria selection) and on the architecture (fast preview mechanism). As a matter of fact not only does the interface play a big role, the software architecture is strongly influenced by the recommendation technique which is used. Pure content-based techniques require too much processing and cannot be processed locally in a client. The repartition of filter discovery and music selection in different components as suggested in MobiXpl is a good lead that we are going to follow in the next section.

## 5.4 OUR PROTOTYPE: THE CONSTRAINT SATISFACTION RADIO AND THE FUMES USER INTERFACE

A prototype called FUMES CSR taking into account the aforementioned concepts has been designed, implemented and tested. The realisation of the prototype and part of its evaluation have been achieved in collaboration with two other students: Sascha Gebhardt from the Ludwig-Maximilian University (Master thesis on the design of the FUME interface) of Munich and Clemens Hahn from the University of Ulm (Diplom thesis on the design of the CSR Backend).

### 5.4.1 *Technical requirements and the logical architecture of the personal radio*

In order to validate the architecture that we introduced in the previous chapter, we had to reuse a significant part of the concepts that we develop in the previous chapter. However some technical requirements that we introduced in order to be as close as possible to a serial vehicle in the design of the former prototype were not relevant. It rapidly turned out that the existing user interface had to be completely re-designed and re-developed. Moreover, in order to achieve reactivity within the system, we had to extend the existing client server architecture, with new components that we are about to detail.

#### 5.4.1.1 *Functional requirements*

The functional requirements derive from the use cases that we have formerly described and the comparative study that we have lead over the state of the art. The criteria which are used for the creation of the playlist take into account the user relevance as well as mobile availability of metadata considerations (see chapter 3).

- Enable the creation of music playlists using combination of multiple criteria
- Provide a discovery process for the query formulation
- Provide a fast preview system in order to diminish long cognitive workload
- Give the possibility for the user to select alternative results without reformulating the query

Furthermore, we have formerly (*cf. supra p.*127) given a list of good practices for the design vehicle user interfaces in IVI systems.

### 5.4.1.2 *Not functional requirements*

- the music has to come from (an) online catalogue(s)
- the skipping behavior has to be taken into account for music suggestions
- the metadata has to be aggregated from different sources

## 5.4.2 *The user interface a new input methodology*

The interface of the FUMES CSR prototype is split in two parts [Turlier *et al.*, 2010, Gebhardt, 2010]: the playlist creation mode where the user and the playlist consumption mode. The user can switch between the two modes at any moment. For the playlist creation mode, the user algorithm for a playlist creation is the following:

```
createPlaylist = false
while not createPlaylist
        display criteriaCategories
        select criteriaCategory
        ask changeOrremove
        if criteriaCategory change is true
                display criteriaCategoryItems
                        select item
        endif
        if criteria remove
                remove criteria
        endif
        compute preview
        ask createPlaylist
done
```

### 5.4.2.1 *Criteria selection*

In order to give the user the possibility to personalise his playlist, we have decided to base the user interface on category filtering since, free-text search has too many shortcomings for the use case that we target. Instead, we have implemented the fuzzy multi-criteria e-search (FUMES [Gebhardt, 2010]) which allows the user to define playlist filters which are then processed by the constraint satisfaction radio components (CSR [Hahn, 2010]).

Figure 47: Fish-eye menu for the discovery of the criteria

Figure 48: Selection of the mood

FISH-EYE TREE-STRUCTURED MENUS For the selection of criteria among long lists (like the genre list, or the time selection), it is necessary to help the driver to identify what is the item current in focus. A menu list with fish-eye effect consists in giving a bigger font size to the items in focus and to decrease the size for the items distant from the focus (see 47). In order to make the display of the items relatively stable and avoid obtrusive change of the items size when scrolling [Bederson, 2000], it is possible to use the combination of a linear and a spherical function [Gebhardt, 2010].

MOOD SELECTION We have seen that mood can be used as a criteria to select a playlist. However, it has to be defined using the combination of two orthogonal values (valence and arousal). We have also seen that, the representation of this input like in Musicovery (see figure 39) is not suitable for a vehicle user. Therefore, we have mapped the 2D representation of the moods in an horizontal list with a preview of the tracks that corresponds to the focused item 48.

### 5.4.2.2 *Combination of different criteria and preview*

An important aspect of the personalisation is the combination of the different criteria. The user can select and remove criteria in the playlist creation panel. Once a criteria is added and removed, a preview is displayed almost instantaneously so that the user can make a decision whether he needs to refine the query or if he should start consuming the playlist (see figure 49). The following criteria can be chosen by the user:

- The genre: The user can chose in a list of 542 different genres. This list is organised in a tree structure with a first level of 15 genres in order to facilitate the navigation
- The mood: We compressed the 100 different mood description to 25 and linked their 2D distribution to a 1D list

Popularity
Hot
Genre
Alt/Punk
Mood
Brooding

Figure 49: Combination of different criteria and preview

Popularity
Hot
Genre
Alt/Punk
Mood
Brooding

♪ Take Me Out
Franz Ferdinand
Franz Ferdinand
Indie/Alternative
2004 3:57

Figure 50: Playback mode of the FUME CSR prototype

- The popularity: We distinguish three levels of popularity by combining social and crawling data: Underground, Mainstream and Rock
- The time period: The user can define a precise year, add some variance or
- The source: The music may come from a personal library or from an online catalogue

The user can select several items within the same category, they are then interpreted as a disjunction (for instance select tracks such that `genre is Jazz OR genre is Rock`. When filter items belong to different categories, they are interpreted as a conjunction (for example select tracks such that `genre is Rock AND popularity is Hot`).

### 5.4.2.3 *Consumption panel and playback*

The playback panel is displayed when the user requests the creation of the playlist based on the criteria that he has defined. The playlist is displayed as a chain of tracks represented by album covers icons. The goal is to reproduce a radio-like listening behaviour which is logical to the understanding of the user.

HISTORY The history of the tracks formerly played back, is displayed on the left of the panel. The track currently played is displayed in the middle.

ALTERNATIVES The right of the playback panel is split in three chains of music tracks: (1) in the middle, the next tracks of the current playlist which meet the filter which has been created by the user, (2) on the top, a list of tracks belonging to the same album as the track which is currently being played and (3) on the bottom, an alternative playlist which is based on social based recommendation for the track which is currently being rendered. Using the iDrive functionalities that we have presented in figure 11, the user can switch between the alternatives by shifting the iDrive, select by pressing it and browse suggestions by rolling it [Gebhardt, 2010]. This display of the different alternatives and of the history in a same panel, gives to the user the possibility to modify the playlist whilst listening to it and saves user interactions.

### 5.4.3 *Logical components of the personal radio*

Some of the logical components that we have formerly defined in chapter 4 can be reused (simplified vehicle client, server-vehicle communication interface and abstraction of music services). Nevertheless, many new components were added to the logical architecture in order to meet the new functional requirements.

#### 5.4.3.1 *Vehicle client*

The vehicle client does not have all the functional requirements of the previous chapter. We did not target an offline use of the client, therefore no complex media storage and indexing are provided in this personalisation scenario. The focus has been given to the development of the user interface. The media rendering, only support audio formats. The media resource resolution and rendering components remain though the same.

#### 5.4.3.2 *Abstraction of services*

The abstraction entails much more services than for the personal radio, not only do we integrate data from the music data services but we also use metadata services and social services. When the user interface loads, all the semantic structures of filter categories are retrieved from the caching component so that the user can directly start creating a filter for the playlist he wants to listen. This step avoids the client having to connect to each of the service in order to discover what are the filter parameters.

MUSIC DATA SERVICES The music data service is the same as for personal radio, with the distinction that we focused on only one music provider (Rhapsody). We can do that without loss of generality since we reuse the same abstraction layer as for personal radio.

METADATA SERVICES External metadata services are necessary for the retrieval of the following information: music popularity and music mood since they are not delivered together with the music data.

SOCIAL SERVICES We also appended an abstraction layer of social services in order to implement implicit (love/hate) and explicit (scrobbling) relevance feedback functionalities.

#### 5.4.3.3 *Aggregation*

The aggregation component is a central functionality of our solution (see also [Hahn, 2010]) since we aim at using different sources of metainformation in order to create the playlist. Those sources are accessed through the aforementioned abstraction layer to maintain a provider agnostic design. In order to optimise the use of the visual

| | seed song 1: Mando Diao - Gloria | | | seed song 2: Kasabian - Fast Fuse | |
|---|---|---|---|---|---|
| T1,1 | Mando Diao | High Heels | T1,2 | Kasabian | Take Aim |
| T2,1 | The Libertines | Can't Stand Me Now | T2,2 | Arctic Monkeys | Fire and the Thud |
| T3,1 | Johnossi | Man Must Dance | C4 | **Editors** | **Munich** |
| C1 | **The Kooks** | **Do You Wanna** | T3,2 | White Lies | Death |
| T4,1 | Sugarplum Fairy | She | T4,2 | Franz Ferdinand | Turn It On |
| T5,1 | The Hives | Walk Idiot Walk | T5,2 | Arctic Monkeys | Potion Approaching |
| C2 | **The Hives** | **Tick Tick Boom** | C1 | **The Kooks** | **Do You Wanna** |
| T6,1 | Johnossi | 18 Karat Gold | T6,2 | The Libertines | Can't Stand Me Now |
| T7,1 | Razorlight | Wire To Wire | C3 | **Razorlight** | **America** |
| C3 | **Razorlight** | **America** | C2 | **The Hives** | **Tick Tick Boom** |
| C4 | **Editors** | **Munich** | T7,2 | Kaiser Chiefs | The Angry Mob |

Table 14: Similar tracks for two seed songs, collected from Last.fm's web service (see also Hahn *et al.* [2010])

modality, we need to provide the user with rich (optimal visual load) and consistent (frustration avoidance) information: text and images. This component can acquire semantic information from different service providers, link them and process them simultaneously. We have seen in section 3.4.2.3 techniques to reconciliate inconsistent data. This component implements them in order to deliver consistent data to be stored and cached by the caching component, and to be displayed by the user interface.

#### 5.4.3.4 *Caching*

[11] The reactivity of the application is a key requirement for the success of the user evaluation. We have seen in chapter 3 that third-party providers have very different qualities of service, especially in terms of latency. Therefore, it is not thinkable, to just access them concurrently and aggregate their results when trying to resolve a query. We implemented a caching mechanism, where a representative sample of the online dataset is stored and periodically updated.

PREVIEW COMPUTATION The role of the caching mechanism is to process quickly the request of the client with local data, without accessing any third-party. A list of songs which meet the requirements is filtered out from the local data and used for two purposes: (1) display a preview of the current filter choice and (2) have a basis of 'seed songs' in order to compute the whole playlist.

CREATION OF THE PLAYLIST The playlist generation process is based on the preview tracks which consist in a list of $n$ seed songs; $\mathcal{S}[0] = \{s_1, s_2, ..., s_n\}$. For every $s_k$ there is set of $m_k$ recommendations $\mathcal{R}_{s_k} = \{t_{1,k}, t_{2,k}, ..., t_{m_k,k}\}$ which is retrieved from metadata providers. Our algorithm process incrementally and takes the first seed song $s_1$ and searches in $s_2, ..., s_n$ a song $s_k$ such that the cardinal of $\mathcal{C}_{s_1,s_k} = \mathcal{R}_{s_1} \cap \mathcal{R}_{s_k}$ is maximum, that is to say that the recommendations of $s_1$ and $s_k$ have the maximum of tracks in common (see figure 51-1). The algorithm then carries on with the set $\mathcal{S}[1] = \{s_k, s_2, ..., s_{k-1}, s_{k+1}, ..., s_n\}$ where $s_1$ has been removed, until $\mathcal{S}[n-1]$ when the set of seed songs

---

[11] Text and illustrations of sections 5.4.3.4 and 5.4.4 have been co-published in Hahn *et al.* [2010]

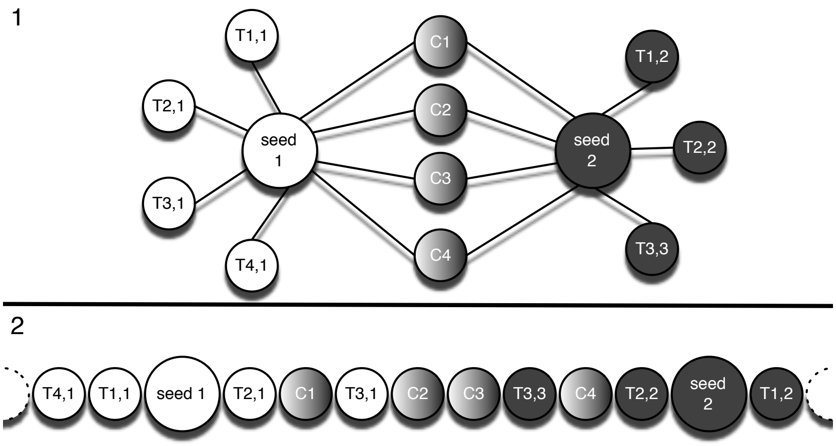


Figure 51: The interlacing method to generate playlists with smooth track-to-track intersections. At the top two seed-songs with similar tracks (T) and common similar tracks (C). At the bottom the generated playlist with interlaced tracks for two seed-songs.

is exhausted. This way, we create an ordered listed chain of seed-songs $\mathcal{S}' = \{s_1, s'_2 ..., s'_n\}$ where $s'_k$ are a permutation of $s_k$ and sets of common songs which can have different cardinality $\mathcal{C}_{s_i, s_j}$. A playlist can be created by placing the common songs between the seed songs as following: $s'_i$, $\mathcal{C}_{s'_{i+1}, s'_i} \setminus \mathcal{C}_{s'_i, s'_{i-1}}$, $s'_{i+1}$, $\mathcal{C}_{s'_{i+2}, s'_{i+1}} \setminus \mathcal{C}_{s'_{i+1}, s'_i}$ where $\mathcal{C}_x \setminus \mathcal{C}_y$ is the difference between sets $C_x$ and $C_y$.

The figure 51-2 and the table 14 illustrate how the remaining tracks that are not common to seed songs (*i.e* they belong to the complementary of $\mathcal{C}_{s'_i, s'_{i+1}}$ in $\mathcal{R}_{s'_i} \cup \mathcal{R}_{s'_{i+1}}$) are interlaced between the common tracks completing the result, in order to create a playlist smoothly moving from a seed song to another. In order to deliver a playlist to the user as fast as possible, the collections of similar songs are (a) parallel retrieved and (b) the playlist generation is split in multiple parts. The parallel request for similar tracks accelerates the generation process. Depending on the latency of the service providers, waiting for the response takes a significant amount of time in generating the playlist. By splitting up the playlist a first part of it can be delivered in an acceptable delay to the user. Thereupon, whilst listening to the first tracks of the playlist, the other parts can be built up in background.

### 5.4.4 *Technical architecture and specific implementation*

The software for the aforementioned concepts has been developed and integrated in a BMW vehicle in order to be tested (see chapter 7).

#### 5.4.4.1 *The vehicle client*

The vehicle client has been developed in Flash Action Script 3 and runs in a specific web browser that had been developed for research purpose at BMW Research and Technology. This web browser is based on the Webkit HTML engine[12] and can be linked to a CAN interface in order to translate the messages from the knob (the iDrive) into DOM mouse events. A javascript mapping script is integrated to the HMTL

---

[12] www.webkit.org

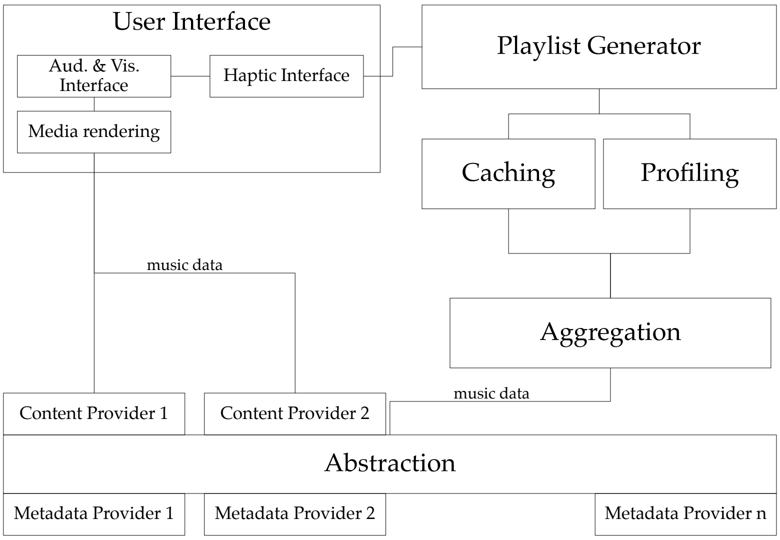


Figure 52: Architecture of the client

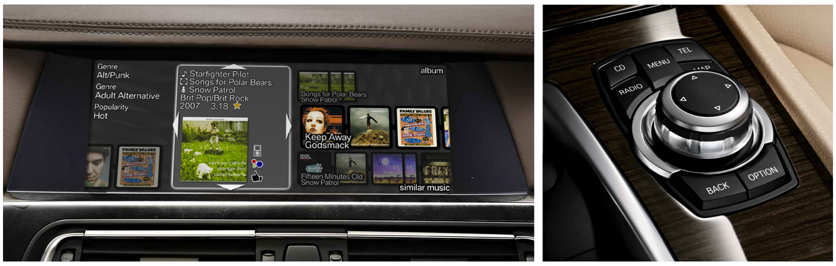


Figure 53: The experimental vehicle. left: graphical user interface, right: controller knob

page which loads the flash applet in order to forward mouseevents from the DOM to the applet. The browser runs on windows PC located in the coffer of a 7 series and whose DVI output is transcoded to a LVDS output for the CID display (see figure 53). The windows PC is connected to internet using a UMTS router with a single antenna. We could use this vehicle setup in order to conduct a user study presented in section 7.1.4 that assessed excellent user acceptance results.

### 5.4.5 *The backend*

In order to reuse part of the infrastructure of personal radio, the backend is a J2EE application server (Glassfish) which runs the whole business logic of the aforementioned functional components and of a relational database (MySQL) which assures the persistency of the caching and of the user profiles. The database is accessed through an abstraction layer (Hibernate). Objects of the business logic (User profiles, music tracks, music playslists, *etc.*) can be directly mapped to entity relations in the database. The backend was connected to the internet over a 4MBit/s bidirectional DSL connection. The communication between the backend and the server has been slightly modified in comparison with the previous chapter, since the client requests perform other URLs instead of XML messages (SOAP). This makes the fast debugging with any web browser much more simple, and allows to write scripts very easily for the testing of the architecture.

CACHED DATA In order to deliver fast preview to the user for all sorts of queries, a representative sample of the metadata of the online catalogue has been created and enriched with metadata from other providers as following [Hahn, 2010]: For each of the 542 genres which are available, up to 50 of the most popular tracks of each genre were stored; which results in a list of 19500 tracks of 5300 different artists and from 9650 different albums. We added to them a list of 3795 tracks for which Gracenote provided us with mood metadata, and a list of the 1000 popular tracks according to the Echonest. Additional metadata information was retrieved for the resulting dataset from The Echo Nest and from Last.fm for the *hotttness*, *familiarity*, *playcount* and *listeners*. We measured the efficiency of the arhitecture under some stress simulation and evaluated the representativity of the cached data in section 7.1.6.

## 5.5 CONCLUSION

More can be less in vehicle infotainment, if proper decisions are not taken into account when designing the system. On the one hand the media on-demand architecture that we introduced earlier bestows the user a tremendous quantity of media content and means more freedom and 'long tail' reach. On the other hand active choice and personalisation entails a proper usage of the different modalities to avoid resource competition in the cognitive load with primary and secondary tasks. The user interface must be reactive, easy to understand and still provide enough liberty in its tuning and tweaking possibilities. This is only possible in leaving the most resource consumming tasks of the playlist creation process such as the metadata aggregation and the feature filtering in a remote component which can process them while delivering pre-cached information for an immediat display to the client. We present in chapter 7 a user evaluation of the prototype that we have made. We have presented a music browser solution which covers both Yet, we have not completely covered the personalisation dilemma. As Seufert and Ehrenberg [2007] noticed, "more or less stable preferences for certain types of media or content categories are 'hidden' behind unstable fluctuations in the acute media use behavior caused by short-terms context variable". As we will see in the following chapter, the personalisation does not always have to be actively trigerred by the user.

# 6

# THE CONTEXT-DRIVEN USER-PASSIVE PERSONALISATION

> *" A robot may not harm a human being, or, through inaction, allow a human being to come to harm.*
> *A robot must obey the orders given to it by the human beings, except where such orders would conflict with the First Law.*
> *A robot must protect its own existence, as long as such protection does not conflict the First or Second Law."*
> — Isaac Asimov, *Three Laws of Robotics, 1940*

This chapter presents the context-driven personalisation as a solution to the issues that we have identified in the two previous chapters: (1) reducing the need of a user input when creating playlist filters and (2) improve the relevance of localised advertising. The chapter first gives a thorough analysis of the functional requirements and architectural requirements of a context driven system in terms of context model, context reasoning method and integration of the context reasoning in the infotainment architecture. Aftewards, the chapter describes the Vehicle Program Director Framework which is a prototyped study illustrating the solution of the passive personalisation problem.

## Contents

## 6.1 POSITION OF THE PROBLEM

Asking the user himself to formulate a query in order to access some media content is not always possible. Sometimes the cognitive load is too important, and the driver should not be disturbed, the task of formulating the same search query every day can be seen as tedious and could lead the user to avoid using the infotainment functionality. Further more, as we previously mentioned, even when the user is willing to create a playlist, the number of criteria to select can be discouraging.

As suggested by behaviorists (cf. infra section 2.3 *p*.67) using the context so as to filter content, adapting this content and, automatically creating playlists is a solution to be considered in order to simplify the use of infotainment systems (generate more easily a personalised program) and to increase the serendipity (find relevant information that the user may not know) of infotainment programs proposed to the user.

Actually, the idea to use the context in order to help the user is a long-lasting idea in the field of autonomic-computing. On the one hand different research prototypes have been designed in order to validate architectural concepts: like centralised or distributed selection and adaptation of content and services based on user profile and user context. Kazi-Aoul [2008] or Weiß [2009] focused on architectures capable of adaptation or selection (or even both) of multimedia content based on the context of use. Specific web-services architectures have been proposed by Kaltz and Keidl. Moreover, some research efforts have focused on the problem of context information provisioning.

In our scenario, the use of search criteria is strongly influenced by the context. Indeed, there are already very rich filters to select content and on the other-hand contextual information can be modeled and retrieved using various architectures, however, the contextual retrieval research still falls for short of expectations. This lack of research is probably due to the lack of information about user social patterns and their relations with information consumption. Loeb [1992] stated, “many of the research issues involved in the proper design of high-performance filters are addressed only for a specific and relatively narrow class of sources and users.” In a nutshell, the specialists of indexation can build systems that achieve a tremendous recall and precision, but on the other hand they are difficult to generalize and therefore, we cannot directly bind them to a system which could understand the needs of the user based on its contexts.

As a matter of fact, one of the biggest issue of intelligent systems has been identified by Lieberman [2009]: “Systems that understand the relationship between goals and actions can help users, providing intelligent context-sensitive help.” As we will see, some research efforts tried to dig this intelligence from statistical analysis. According to Lieberman, knowledge that derives from statistical heuristics is not sufficient and needs to be filtered with the so called 'commonsense knowledge'. Unfortunately, common sense is difficult to define formally as we will see.

This chapter is about a the vehicle program director framework that aims at supporting the user in his media selection process. The vehicle program director is a prototyped study of design and integration of context-awareness in the mobile media ecosystem of a vehicle. That means that we are supposed to study three important concepts:

- a context model: The context of the driver of a vehicle.
- an action model: The creation of playlist with different criteria. We have already tackled this problem in the precedent chapter.
- an condition model: The formal definition of a pattern describing the condition defining the action in function of the context, *i.e.* the so called mapping from 'goal to action'. We will further name it the context-reasoning model.

We will first review the very rich literature of context-driven systems, starting from the definition of context made by Dey and then reducing the field to the context of a vehicle. Afterwards we will tackle the problems of reasoning with context data and the persistence of context reasoning knowledge which are very important topics when dealing with automotive systems. Eventually, we will illustrate the concepts that we have developed with a prototype and we will demonstrate how it can be integrated into our infotainment system.

## 6.2 CONTEXT-DRIVEN SYSTEMS

The first systematic study of context-driven systems was done by Schilit [1995] in his PhD Thesis. He focused in his work on the definition of a communication model between different agents capable of reconfiguring according to context information. Schilit [1995] gave a general idea of context resolution between clients entities which consume context information and servers entities which provide it in a dynamic environment, but he did not try to define systematically context adaptation. The most widely accepted definition [Chen and Kotz, 2000] of context-driven systems in the computer science community is the one given by Dey and Abowd [2000] "A system is context-aware if it uses context to provide relevant in- formation and/or services to the user, where relevancy depends on the user's task." This very broad definition covers a wide range of context-driven systems, such as user guides or user service selection.

In the automotive domain, Huber [2008] has provided a general communication architecture for the provision of context information to so called "consumers" that can register to a central server distribution information about context "providers". He also identified attributes of context values that are accordingly relevant to the automotive domain: reliability, precision, freshness, history. Unfortunately, [Huber, 2008] does not provide any systematic methods to evaluate those attributes. However, this approach clearly tends into the direction of a semantization of context data for the automotive domain.

Moreover, in the field of vehicle driving assistance, another prototype architecture had been developed at BMW Research and Technology called the Connected Drive Context Server. It had been designed in

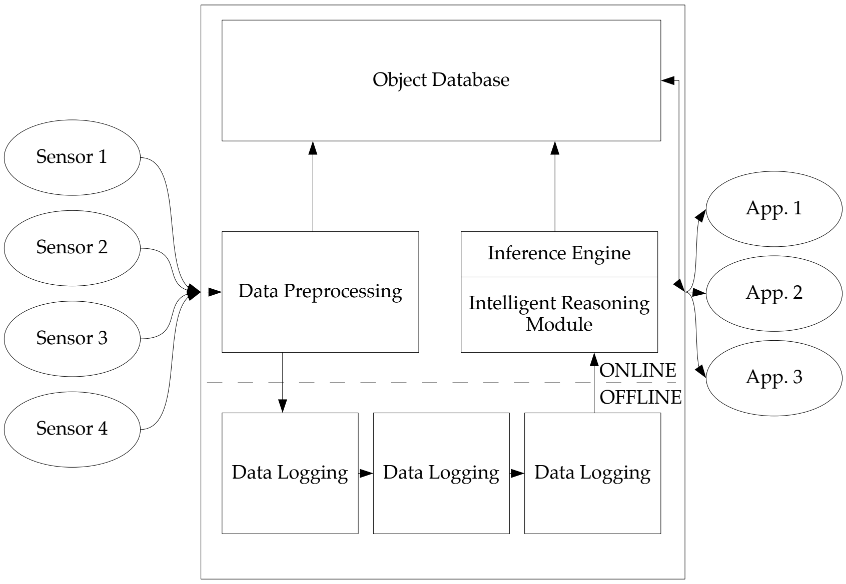


Figure 54: Architecture of the Connected Drive Context Server [Hoch *et al.*, 2007]

order to derive unspecific context knowledge and then to adapt it specifically to different applications (see figure 54), a demonstrator using a camera, line-cross detectors and vehicle detectors had been developed. Even if it is not directly applicable to our problem statement, this architecture illustrates the different logical components that make up a context driven system for a vehicle application:

- A component to aggregate data from the sensoric
- A component to extract knowledge from this sensoric
- A component to compute according to this knowledge the actions that have to be taken

The offline components to extract knowledge from the sensoric suppose a prelearning of the system in order to be able to label the data adequately. Even though if Hoch *et al.* [2007] claim to have create a general architecture. It is difficult to imagine that the learning of the system can be completely application agnostic. Indeed, this system had been designed to infer the intention of the driver to leave his lane and was intended for security purpose. Therefore, the representation model of the context information itself, cannot be completely split from the logic of the application [Bolchini *et al.*, 2007].

This architecture actually describes the logical components. But, their integration in a mobile multimedia architecture is still a challenge. Moreover, Hoch *et al.* [2007] left opened the problem of reasoning that we have stated before; *i.e.* how can the system decide the decision to be taken in a specific context? Eventually, the notion of the persistence of the knowledge has not been addressed.

In the infotainment domain, a context toolkit has been proposed by the Fraunhofer institute [Zimmermann *et al.*, 2005] that strongly emphasizes on the creation of a semantic layer between the sensor and the control layer (see figure 55). The sensor layer delivers inputs of raw data which cannot be directly interpreted by the control layer which process actions. Between them, the semantic layer first derives semantic context knowledge from the sensor input and then derives upper concepts of context.

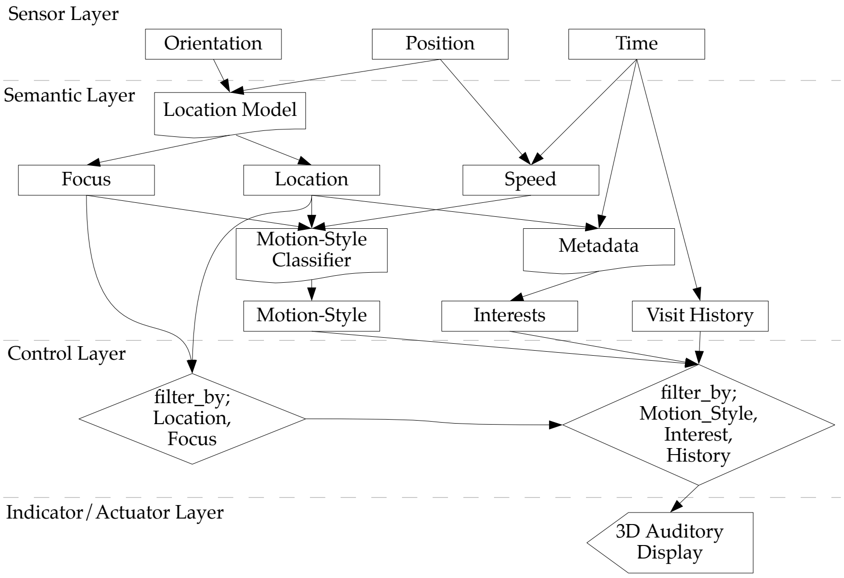


Figure 55: Architecture if the LISTEN prototype developed by Fraunhofer [Zimmermann *et al.*, 2005]

## 6.3 CONTEXT-REASONING FOR INFOTAINMENT APPLICATIONS: A STATE OF THE ART

Most of the aforementioned systems are general thoughts on architectural concepts of a context driven system. They define the different architectural components which are necessary for the realization of context-driven systems. However, they still need to implement context reasoning to be able to process context information. As we will see the context reasoning methods depend heavily of two aspects: the type of context data which are taken in consideration and the type of context-reasoning which is targeted.

### 6.3.1 *The context-awareness model*

The development of context-aware computing has been largely influenced by the activity theory developed by the soviet behaviorist psychologist Vygostky and afterwards extended by the scandinavian Engeström. Computer science researchers as Lieberman who aims at 'mapping goals to actions' are actually referring to the structure of human activity of Engeström (see figure 56). This sociological model cannot be of course directly mapped to a logical model, but is has the advantage to give the insights to make a context-awareness model for a specific application.

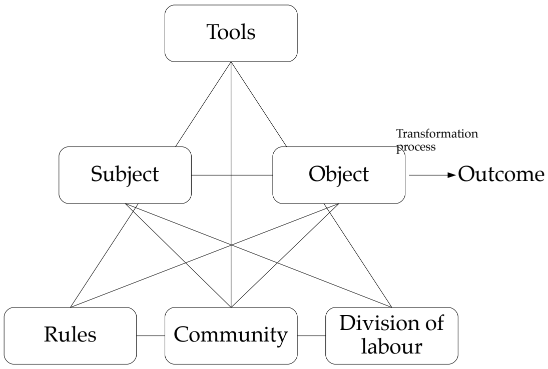


Figure 56: Activity Theory of Engeström

| | Presentation of information | Automatic execution of a service | Tagging of context to information |
|---|---|---|---|
| Discovery | × | | |
| Query Formulation | | × | |
| Filtering | × | × | |
| Result selection | × | | |
| Rendering | × | | |
| Modification | | × | × |

Table 15: Type of contextual action taking for a context-aware infotainment system according to the taxonomy of Dey and Abowd [2000]

1. Subject refers to what Dey designates "identity". The context adaptation is depending of the user characteristics (or a group of users which refers then to the community concept).
2. Object refers the infotainment process that we formerly described. Objects can either be services to be selected in the discovery process (like a search engine with textual input, or an interface with predefined parameters), or media assets.
3. Rules have to be defined, inferred or derived and are the core logic of the adaptation.

Dey and Abowd [2000] extended the taxonomy proposed bySchilit [1995] in a threefold types of context-awareness: "(1) *presentation* of information, (2) automatic *execution* of a service, (3) *tagging* of context to information for later retrieval". For a general vehicle infotainment scenario that we have presented in 2.3, the table analyzes those different types of contextual reasoning.

As we can see in table 15, most of the context-awareness for infotainment systems is about the presentation of information (like the display of point of interests on a map) and the automatic execution of a service (like a search query), tagging of context to information is however somewhat relevant for instance when doing contextual advertising as we will further illustrate.

### 6.3.2 *The context information model for multimedia adaptation*

Identifying the right context information that has an influence on a task is not only a ‘common sense’ problem. For applications like the ’route lane change prevision’, the observation of the user of the lane and of other vehicles is an obvious choice, since the output of the system (respectively ’line change’) and the inputs of the system (respectively: the vehicle position on the route, the vehicle position to other users and the gaze of the driver) are almost physically correlated. Regarding our infotainment problem, the correlation is somewhat more difficult to make. The problem of contextualised media selection and adaptation for mobile devices involves actually both general vehicle context data which are heterogenous and specific media descriptions: “[...] es ist nicht möglich, ein generisches Verfahren zur Verfügung zu stellen, das alle verfügbaren Kontextinformationen vergleichen kann” [Weiß, 2009, p. 111] [1].

An attempt of systematic categorisation of context information in the field of multimedia adaptation has been made by Kaltz [2006] who developed a methodology for adaptive, context-aware applications. It quite heavily relies on the taxonomy of Dey and Abowd [2000] and lists four categories: “(1) User & Role (2) Process & Task (3) Location, (4) Time and (5) Hardware Device”. The differentiation between spacial and time information is a common aspect of many context-aware systems, as well as the notion of “identity” which is captured by the “User & Role” category of Kaltz. But, it is not quite clear how this taxonomy would define the motion information like velocity or acceleration. As a matter of fact, when designing context-aware system we are confronted to the following problems:

- Application specificity: a context information is intended to be used by the application with a specific logic (for instance using the velocity of the vehicle to decide the rhythm of the music to be displayed does not require the same logic as using the GPS position of the vehicle to find out a contextualised advertisement.)

- Sensoric specificity: sensors deliver values that they are capable of measuring (GPS position, time of the day). Those values my have different types of scale: nominal data, ordinal data, intervall data and proportional data.

- Derivation specificity: contextual information from the sensoric is not always directly usable and needs to be derived from the sensoric values. This derivation generates uncertainty but moreover is to some extent highly dependant of the logic of the application. For instance, from the GPS position can be derived contextual information like the name of the place, or the weather conditions (if available from a remote resource)

We will present in section 6.5.3 an ontological context model which captures data from the sensoric and maps it high-level semantic knowledge.

---

[1] “it is not possible to provide a generic method that compares all context information which is available”

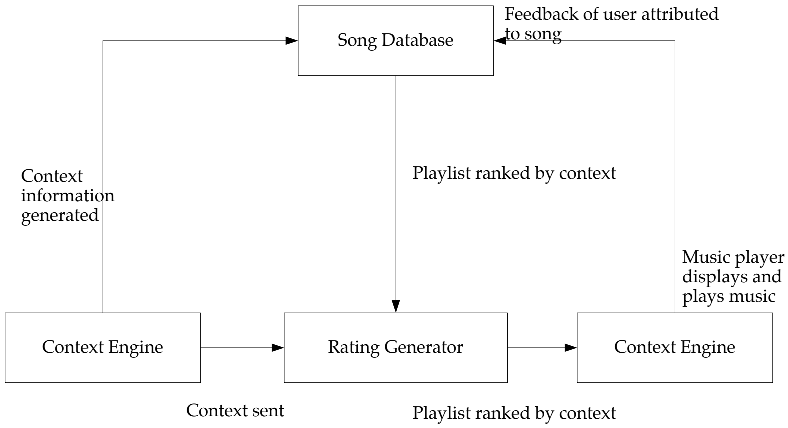


Figure 57: Architecture of Life Track [Reddy and Masciaa, 2006]

### 6.3.3 *Rule based reasoning for information filtering and discovery*

The rule based reasoning relies on logical mappings between a context information and an action to be taken. The rule consists in the logical expression of the knowledge: `IF context_information IS context_type then select Resource`. This knowledge is expressed *a priori* by an expert, a user or the so called 'common sense'.

The following section describes two applications of rule based, context aware information discovery.

#### 6.3.3.1 *Context-tagged media content*

The Life Track [Reddy and Masciaa, 2006] is a prototype that produces music recommendation based on tagged data. In lifetrack context information and infotainment information are processed on the same mobile device. Connection to external services is used only to derive knowledge from the sensoric: get the zip code or the weather situation of the user according to his GPS position.

All the items are tagged with context information. This tagging is done by the user in his personal database which is stored locally. Lifetrack implements a cosinus similarity (see appendix p.215) between a vector with the values describing the context and the values describing the tracks. The calculation made in Lifetrack is quite simple, it consists in a multiplication. Stahl [2003] has systematically studied the similarity functions that can be used (they are summarized in p.215). He makes the distinction between attributes that can be classified which are not hierarchically sorted, and attributes which are hierarchically sorted).

Life Track is a very elegant and simple system (see figure 57) to illustrate the notion of context-defined playlist but it has also some shortcomings. The authors acknowledge that the direct mapping from context to music tracks is not optimal, and they suggest to use high-level concepts like mood and tempo. This direct binding between tracks and context information has also the following drawback, Lifetrack makes only use of the personal library of the user which means that the user cannot use the recommendation with the large online databases. Eventually the authors suggest to make collaborative tagging, *i.e.* to compare user profiles and to derive correlations from it. Nevertheless, this is only possible if users have common tracks in their libraries.

#### 6.3.3.2 *Ontologies and description logics*

In order to generalize the problem of rule-based media content selection, the use of a formal relation between a context model and an infotainment model is necessary. [Kaltz, 2006] introduced a meta-model based on ontologies in order to design context-driven web-services adaptation. It consists in defining different ontologies of concepts describing the context and mapping them with an ontology describing the different domain of adaptation of the application (*i.e.* the selection process, the user interface, *etc.*). A prototype with a similar philosophy has been presented by Weißenberg *et al.* [2006], with FLAME, a system for public contextualized information for visitors of the olympic games in Beijing equipped with handset devices. FLAME aims at providing an architecture to select and push information services about the sport events, based on the context of the users. FLAME defines different ontologies: on the one hand the different domains of information (an ontology of sport information, an ontology of tourist information, *etc.*); and on the other group of ontologies describing the tasks, *i.e.* the services (type of service, parameters) and the situations which are complex combinations of context concepts that capture high-level situations. For instance, the subject service `Watching Competition` is logically associated to the objects of the user position `Stadium`, action and state through the predicate `Being at event`.

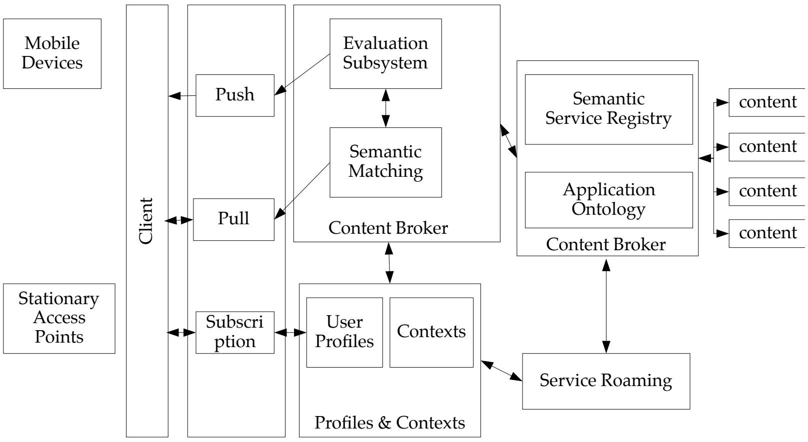


Figure 58: Architecture of FLAME [Weißenberg *et al.*, 2006]

Weißenberg *et al.* [2006] use an OWL reasoner (OntoBroker[2]) to infer service selection based on preregistered situations. They have tested the possibility to store logical rules in an ontology and to process them. The situation description through ontologies enables multiple-

---

[2] http://ontobroker.semanticweb.org/

dimension adaptation and different level of granularity. They identified performance issues and came to the conclusion that current reasoners are not yet as performant as relational databases. Therefore, they had to replace the original 'subject, predicate, object' structure of their tasks ontology, in an entity-relation schema, in order to implement mappings between situation concepts and service descriptions.

This ontology approach gives us interesting insights regarding the possibility of describing formal situation patterns which are the combination of several context information. Moreover, predicate logic can capture logical patterns, mapping situation to service description. However, the prototype of Weißenberg *et al.* [2006] assume that the situations are manually defined and it is not capable of reading any raw sensoric data. Regarding, this aspect, Kaltz [2006] gave an important contribution by introducing the notion of 'degree of activation'. Concepts of an ontology allow predicate logic between heterogen concept spaces (context information and infotainment description) but the reality is composed of physical values which have to be mapped to those concepts. By measuring the degree of activation of context concept, it is possible to calculate the degree of truth of a situation and then decide if action related to it are relevant or not. We will further introduce how the proposition of Kaltz can be extended to fuzzy controllers so that not only the the context concepts can be activated, but also the preferences of the user.

As we can see, the advantage of rule based methods is that the user profiling rules are expressed on a semantic level [von Hessling, 2004]. Therefore, they are human-readable and can be compared on this semantic level. However, as we previously mentioned, the expression of those rules derives from a knowledge *a priori*, *i.e.* expert knowledge or user-defined knowledge. If no such knowledge exists (for instance: Is there a psychological pattern that maps the rhythm of the music to the temperature outside?), then the rule based-method is not sufficient.

Another advantage of this method is that logical rules are persistent, that is to say that they can be stored and replicated in different entities.

### 6.3.4 *Statistical reasoning*

Oppositely, it is possible to consider that the rules mapping context information to a decision can be learned automatically. Some research efforts have tried to derive mappings between contextual information and infotainment description from the statistical analysis of user patterns, trying to find out correlations between user environment and infotainment consumption habits.

In the music domain, a pretty straight-forward approach was led by Omojokun *et al.* [2008]: They let four users listening to music with their laptops as they normally do and recorded the following information: Time of the day, day of week, weather (temperature, humidity, weather description in text and precipitation), user activity (processes running), bluetooth devices in range, noise level in the room in which the user was listening to music, system and application volume level. The information gathered was compared to a signal analysis of the music that had been listened to and based on seventy-height different signal properties.

Omojokun *et al.* [2008] introduced the idea that there could be some pattern linking music signal properties and the user environmental conditions when listening to it. Their results are unfortunately difficult to interpret: They admit that they could no extract any general pattern covering the for users of their study. Their conclusion that only user-specific patterns can be derived from this methodic is quite disputable since their test-sample is statistically too limited.

#### 6.3.4.1 *Neural Networks*

The XPod [Dornbush *et al.*, 2007] is a prototype that creates dynamically a playlist by predicting the chance of a track being skipped by the user. This chance is calculated using a neural network with a fully connected feed forward network with two hidden-layers. On the first layer of the network, kinetic and kinematic information (trans-longitudinal and longitudinal acceleration) as well as physiological information (galvanic skin response, skin temperature) and context information (time of day, and day of week) were used and forwarded to the hidden layers. The outcome is a value that says the chance of the track being skipped (the bigger is the value, the greater is the chance). The error to train the network is computed using the skipping behavior of the user (if the user skips a song, then an error is propagated in the network and the weight of the nodes are updated). The Xpod works as a client-server where the client aggregates the different context information and user feedback and send them to a server that computes the next track to be played.

The XPod is an excellent exemple of a fully integrated learning context-aware system. The action model (the skipping of the user) is directly mapped to a well-known mathematical model (the error function used for the training of a neural network). It has however the following drawback: it relies a lot on signal processing of the musical data which involves scaling problems when dealing with online catalogues. Moreover, it is only capable of recommending music because the descriptors that it uses are intended for a music use. Eventually, the weights of the neural network are computing under an unsupervised learning procedure and are user specific.

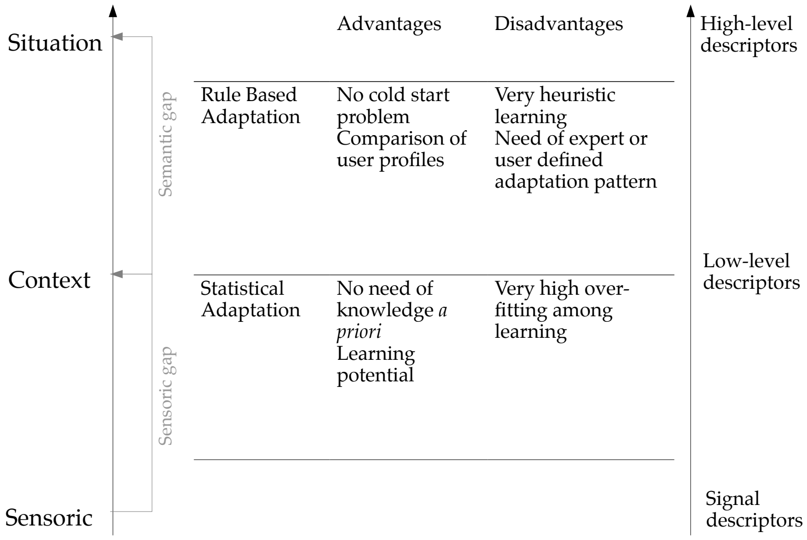
In some other domains (mainly geolocalised recommendation), other statistical heuristics have been applied like Bayesian Networks in [Yuan and Tsao, 2003] for the recommendation of restaurants based on different types of user contexts and ANFIS Neural Networks [Park *et al.*, 2007] for contextual advertising.

### 6.3.5 *Evaluation of the different solutions of context reasoning*

The table 16 compares the different techniques that we have formerly described. As we can see, the different methods differentiate themselves mainly on the source of adaptation logic (which refers to the statistical heuristics that Lieberman opposes to common sense), the possibility of the system to learn and the nature of the adaptation's persistence. Black-box models (such as Neural Network) offer high-learning potential but are difficult to extrapolate and give solutions to local problems whereas white-box models offer the possibility to analyse the reasoning on a semantic level. Moreover, it turns-out that type of adaptation is tightly correlated to the characteristics of the context (raw sensoric or high-level situation) and to the sort of multimedia content to adapt. The

| | Life Track [Reddy and Masciaa, 2006] | Flame [Weißenberg *et al.*, 2006] | XPod [Dornbush *et al.*, 2007] |
|---|---|---|---|
| Reasoning application | Select music track | find information services | select music track |
| Action model | Track selection | Information Service selections | Track selection |
| Context information | Kinetic (acceleration), weather, GPS (zip codes), entropic (noise) | User, GPS position | Kinetic, kinematic, physiological data |
| Infotainment Information | Semantic tags on music tracks | Semantic classes (ontology) of services and context | Low and high level musical information |
| Reasoning method | Cosine similarity | Tableau Algorithm (logical inference) | Neural-network activation |
| Reasoning Architecture | monolithic (client sensoric and web aggregated information) | context information distributed between client and servers, reasoning centralised | client (sensor aggregation)/server (sensor fusion context reasoning) |
| User learning | love/hate feedback; context priority | no user feedback | online learning through skipping behavior |

Table 16: Comparison of different methods of context-aware media and service selection techniques



| | Advantages | Disadvantages |
|---|---|---|
| Rule Based Adaptation | No cold start problem<br>Comparison of user profiles | Very heuristic learning<br>Need of expert or user defined adaptation pattern |
| Statistical Adaptation | No need of knowledge *a priori*<br>Learning potential | Very high over-fitting among learning |

Figure 59: Comparison of reasoning methods for context-based infotainment selection

figure 59 illustrates the different dimensions that have to be taken into account when comparing reasoner systems.

## 6.4 INTEGRATION OF CONTEXT-DRIVEN SYSTEM WITH THE MEDIA DELIVERY ARCHITECTURE

The aforementioned concepts do not only differ in the type of reasoning that they use; they have also very different architectures. This is for our vehicle scenario a very important aspect since as we have already mentioned, the vehicle media consumption architecture has hard constraints in terms of computation resources and network resources.

### 6.4.1 *Distributed and composite architectures*

A very hollistic study of computer architectures for context-driven personalisation of multimedia content for mobile devices has been done by Weiß in her doctoral dissertation [Weiß, 2009], where she systematically analysed the collaboration between content recommendation and content adaptation logical components in a client-server architecture.

Weiß suggests to split the content selection process between the client and the server (see figure 60), arguing that some context-data are updated so often that they would overload the network resources if the client had to update them continuously. The process proposed by Weiß is twofold: first the server selects content according to the profile of the user, and then the client reprocess some filtering depending on the context information locally available. This has the following consequences, the profile of the user (that is to say the user-specific rules for the context adaptation) has to be replicated between the mobile client and the server. In the framework which she describes,the adaptation rules written in an extended version of MPEG-21 (called MASL for Multimedia Adaptation and Selection Language) are synchronized

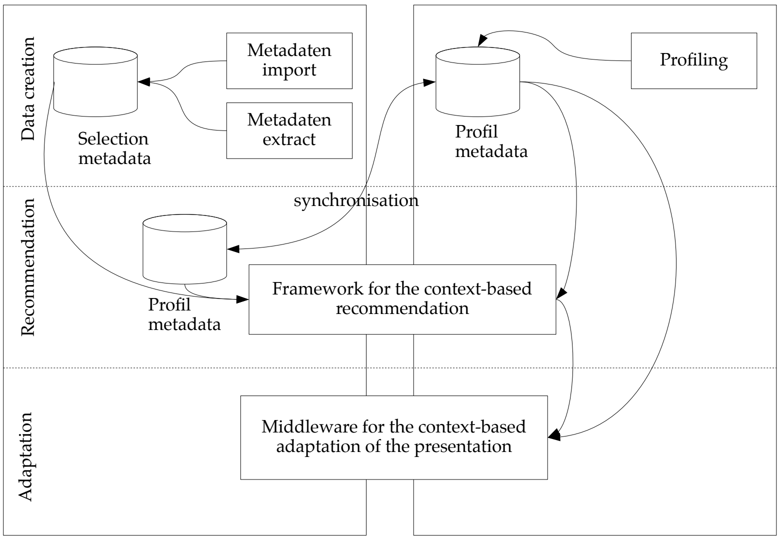


Figure 60: Distributed architecture with generic middleware for client and server [Weiß, 2009, p. 193]

(over a protocol that she does not describe) between the client and the server.

The idea to split the context processing between the client and the server is interesting because it enables some pre-processing and thus save computing resources on the client, and perhaps network resources. Unfortunately, Weiß [2009] does not explain how to concretely achieve the recommendation goal, when only partial information is availabe. This technique has theoretically the serious drawback of making almost impossible any complex situation inference, since context information is not aggregated in a central node. Moreover, as she deploys her context-middleware among both client and server, she makes a weak assumption regarding the computing and networking capabilities of those two nodes which have in practice very different characteristics.

Actually, the architectural discussion of context-adaptation and context-selection relies more in the way it is integrated to existing multimedia architectures. Kazi-Aoul suggests in her doctoral dissertation [Kazi-Aoul, 2008] to split the adaptation policy generation and the adaptation processing (processing of the policy formerly generated, based on the business logic of the multimedia application) between different entities. Her work focused on the implementation of a composite content adaptation platform, which can adapt content to the context of the terminal of the user.

Thereby, Kazi-Aoul can propose a very scalable architecture where the client can switch between adapters independently from the content which have to be adapted. Actually, the main reason why Kazi-Aoul claims that her architecture is so efficient is because it targets an adaptation use case, where the quantity of context information is small and therefore, the complexity of the rules describing the adaptation remains acceptable. More-over, Kazi-Aoul leaves to the user or to a planification entity, the task of choosing the adapter, which means that no personalised context-adaptation is driven by the architecture. Indeed, such a system would need a very important synchronization

| | PAAM[Kazi-Aoul, 2008] | MASL[Weiß, 2009] | CATWALK[Kaltz, 2006] | Service Globe[Keidl, 2004] |
|---|---|---|---|---|
| **Context** | × | × | | × |
| **Situation (complex context association)** | | × | | |
| **Context information aggregation** | client | client and server | client | client |
| **Situation extraction** | none | client and server (limited) | server | none |
| **Logic persistency** | server | client and server | server | server |
| **Adaptation processing** | distributed between peers | server (personalisation), client (contextualisation) | server (proxy adaptation) | server (proxy adaptation) |
| **Evaluation method of the architecture** | quantitative | heuristic and qualitative | case study | none |

Table 17: Comparison of different context-driven media-adaptation architectures

between the adaptation nodes, if user profiles were to be taken into account.

### 6.4.2 *Context proxy web architecture*

Instead of sharing about different entities the context-adaptation of multimedia systems, the Service Globe developed by Keidl [2004] and the CATWALK System proposed by Kaltz [2006] tackle the integration topic of context-aware reasoning in a web architecture with a proxy service between the client and the server. The role of the proxy is to transform the queries of the client and the answers of the server with context-adaptation policies. This architecture has the advantage to reuse completely the existing web architecture.

### 6.4.3 *Summary and comparison of the different architectures*

The table 17 compares the different architectures that we have presented. It turns out that since web architectures are *de facto* centralised, the context adaptation layer tend to imitate this pattern. However, the distribution of context knowledge among entities (client context, general context, *etc.*), yield a necessary distribution of adaptation and reasoning components. As we can see there is not yet a standardised way to validate context-adaptation architectures.

## 6.5 THE VEHICLE PROGRAM DIRECTOR FRAMEWORK

The Vehicle Program Director has been developed in order to answer to the problems that have been raised by the active personalisation strategies. Based on the analysis of the aforementioned solution, we have been able to build a prototype which emphasis on those characteristics:

- Reuse as much as possible the personal radio architecture designed for active personalization
- Consider all types of context information available in-board as well as off-board
- Propose a reasoning method that combines expert knowledge and user learning

### 6.5.1 *The use cases*

CREATION OF A CONTEXT-DRIVEN PROGRAM This use-case is a substitute to the use case that we have proposed in the former chapter. The user does not have to process any playlist query input, since the system generates automatically a personalised channel based on context information. The channel contains the same type of media as those which have been used in the previous chapters, *i.e.* music and podcasts.

CONTEXT-DRIVEN MEDIA SWITCHING FOR ADVERTISING This use-case is an extension of the advertising use case (*cf. infra p.*118). We want the vehicle to decide when it is supposed to display the right advertising. This decision has to be based on the GPS position of the vehicle.

### 6.5.2 *The functional requirements*

A very strong functional requirement of the framework is that it has to be integrated in the existing infrastructure that we have described and developed in chapter 4 and chapter 5. The Program Director Framework targets the following functional requirements:

- Handle the vehicle context information which is available on-board, as well as off-board in order to generate context-driven infotainment programs
- Implement a reasoning mechanismous which handles sensoric data and generates personalization according to the metatada which are available in the vehicle media framework of personal radio
- Implement a mechanism of persistency, in order to be able to save the user context-preferences and to compare them
- Enable learning mechanism in order to improve the generation playlist based on the learning of the user

### 6.5.3 *An ontological model for the context of the vehicle*

As we have seen in sections 6.3 and 6.2, the design of the context model for infotainment applications is influenced by the type of media which has to be selected. The Personal Radio framework that we have formerly presented is designed around high-level semantic descriptions of media assets, especially for podcasts and music. Therefore, the Program Director adaptation reasoning process relies on the mapping of semantically defined context information.

#### 6.5.3.1 *On-board context information*

An important part of the context information can be accessed on-board.

VEHICLE ENVIRONMENT The on-board context information are mainly directly derived from the sensoric information, that is to say from physical values:

- Vehicle cinematic: velocity, acceleration, GPS position (latitude, longitude, altitude)
- Weather context: luminosity, humidity, temperature (outdoor), atmosphere pression
- Habitacle information: temperature (indoor), noise

This physical values are all expressed on interval or proportional scale (for example $\mathrm{km}/\mathrm{h}$, $^\circ\mathrm{C}$)., which enables a lot of mathematical tools from the data mining field to be used [Runkler, 2010].

The driver assistance research has introduced new sorts of sensors which are available in applications like the parking-distance control, or distance to other vehicles. Different technologies (LIDAR, LASER, etc.) can be used, embedded cameras are now being shipped with series vehicle, and there is no doubt that the sensoric is going to boom in the next generations of vehicles. As a result, very intensive research efforts are being lead to realize the most accurate possible model of the vehicle environment, which will lead in an increasing number of information to aggregate.

DRIVE HORIZON The navigation software of today's vehicle is capable of extracting from the GPS position the following information: country, crossing, house, street, town. This information is nominal information (for instance: Germany, Unter den Linden, Berlin), and are stored in local databases.

Another important contextual information, is the intention of the driver. Where is he driving to? For what purpose? The state of the art of driving assistance cannot yet answer globally and systematically to those questions. It is statistically proven that most drivers have six different driving destinations [Ehreke, 2008]. The ILENA project [Patrascu, 2009a,b] at BMW Research and Technology have built a system that can predict the so-called driving horizon based on the first hundred meters of a drive.

#### 6.5.3.2 *Off-board context information*

Off-board context information that cannot be known from the vehicle because they cannot be directly sensed from its environment (type of precipitations: rain, snow, *etc.*), because it needs to aggregate context information from different vehicles to be processed (traffic circulation condition) or because they involve too much storage capacity and need to be updated to often so that they are not directly available. Those context information are based on the aggregation, computation and prediction calculations involving different entities and sources that a vehicle alone cannot access.

In the navigation field, number of off-board solutions are existing in forms of on online databases containing points of interests and social context information.

As we can see, there is redundant context-information between the sensors themselves and between on-board and off-board context sources. Redundancy of information is a good thing because more information can help to validate or invalidate context-driven decisions, but it raises issues because it can make the system inconsistent (what if the outdoor temperature measured by the vehicle is 10 °C and the weather forecast says 20 °C?). The topic of sensor-fusion goes far beyond the scope of this dissertation and will not be addressed any further.

#### 6.5.3.3 *Structuration of context information and expression of complex situations*

The different context information can be organized as we have previously said in an ontology where the different classes represent the context information concepts. But, the ontology alone can only be used to process predicate logic. Raw sensoric data must be mapped to predicates, *i.e.*; algebric values have to instantiate individuals for the classes of the ontology in order to be logically processed by a reasoner.

To achieve this mapping from physical value to the logical concepts, we have decided to use fuzzy logic, in order to express what Kaltz [2006] says to be 'the degree of activation'.

Each concept of the ontology is given some description attributes which define a fuzzy set that can be partitioned into different terms. For instance, the concept `Drive Destination Range` is a variable that can be depicted as following using a 3-term partition: `near`, `local`, `far`. The fuzzy partition of the concept allows to describe a context with different terms at the same time. For instance, the range can be near and local, but with different degrees of truth. To achieve this, each term makes a partition by using a membership function which computes, given a sensoric value the degree of satisfaction of the concept between 0 and 1. Different membership functions have been proposed in the literature [Borgelt *et al.*, 2003]. For the sake of simplicity, we will in the following use trapeze functions. For instance given the distance $x$ we can make a membership function, $\mu^{near}_{Range}(x) \in [0,1]$:

$$\mu^{near}_{Range}(x) = \begin{cases} 1 & \text{if} \quad 0\,\text{km} \leqslant x < 10\,\text{km} \\ -0.1x + 2 & \text{if} \quad 10\,\text{km} \leqslant x < 20\,\text{km} \\ 0 & \text{otherwise} \end{cases}$$

For each of the concept, the membership functions $\mu^{near}_{Range}$, $\mu^{local}_{Range}$ and $\mu^{far}_{Range}$ define the fuzzy set of the `Destination range` class. The figure 61 illustrates the mapping between semantic terms and membership values, for some value of the destination range.

Our ontology can capture various context concepts like the vehicle motion (velocity, acceleration), the time (time of the day: morning,

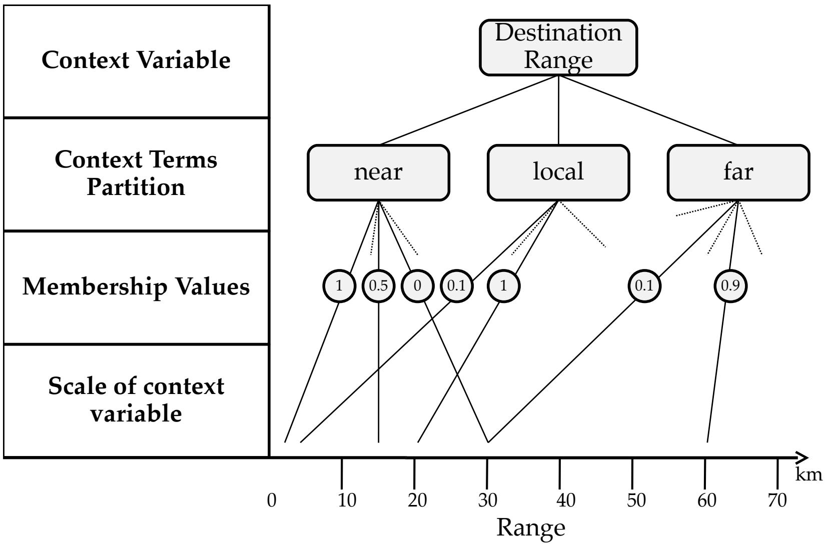


Figure 61: Membership values for an example of context variable: the destination range

lunch, afternoon,...; week part: week-end, worked-week,...; season), traffic conditions, weather conditions (precipitation rate, temperature, humidity, luminosity), landscape (urbanization rate, altitude), destination (range, frequency). A this point it is important to notice that this context model has the two following requirements: (1) It is necessary that a logical partition of the concepts can be made, (2) the formulation of the membership function derives from an expert knowledge. The advantage is that the concepts can be reused in logic assertions as we will see in the next section, the drawback is that some concepts like `Drive` Purpose is Home cannot be directly used, since the different drives purposes (`Home`, `Work`, `Holliday`,:. cannot be mathematically partitioned as such.

Actually, the `Drive Purpose` is an exemple of what we call a complex situation like the notions of `Driving` Style or `Habitacle` Atmosphere; *i.e.* they must be captured using a combination of different context concepts which express a higher level of granularity and thus can be derived from sensoric information.

Complex situations can be approximated using concepts which can be partitioned. For instance, we can assume that the destination `Home` derives from the following predicate `Time` is Evening and (Destination Range is Local or Destination Range is Near) and the destination `Work` derives from the following predicate `Time is Morning and (Range is Local or Range is Near)`.

### 6.5.4 *Expression of user context preferences on a semantic level*

The framework of the program director is user centric and aims at building a context adaptation model as a casual user would describe it. The system must be able to capture and process preferences like "When I am stuck in the traffic jam going to my work, I prefer to listen to chilling music and reportage about science". From this point of view, user preferences are not about deterministic mappings between the value of a sensor to the key, or the beats per minutes of a song.

### 6.5.5 *A remote reasoning model based on a fuzzy controler*

The reasoning model is, as we described, the condition, mapping the goal to be achieved to an action to be executed, that is to say generate an infotainment channel as we announced in the first use case *p.* 166 . We have described in chapters 4 and 5, what are the actions that have to be executed to create personalized infotainment programs using the combination of description metadata.

The personalization of the filtering has to substitute to the user by trying to formulate itself the query. This has the following advantage: our framework can rely on the same filtering system as the one which has been developed for clients with high-interaction possibilities. We already know what is the 'action' to be processed, and we have the tools to process it. Moreover, since both the context model and the infotainment program creation query are based on an ontology, the mapping that we have to describe is a logical mapping.

There are different ways to implement logical mappings. Perhaps, the most obvious way is the use of the existing reasoners which implement the tableau algorithm. However, performance problems have already been mentioned by Weißenberg *et al.* [2006] as we previously noticed, and user learning is difficult to achieve. Spreading activation has been considered by Vallet *et al.* [2007] to activate heterogen concepts in different ontologies.

#### 6.5.5.1 *Fuzzy user preference formulation*

Fuzzy ontologies and fuzzy description logics reasoners are still an active research topic [Bobillo and Straccia, 2008]. We have investigated a third sort of logical mapping, based on fuzzy controllers. We defined a semantic infotainment preference category which is a logical concept that can be partitioned in three values: `Important` (assets of this category must present in the playlist), `Secondary` (assets of this category can be present in the playlist) and `Negative` (assets of this category must not be present in the playlist). The preference can take a value between -50 and +100, the figure 63-1 illustrates the corresponding fuzzy-set using sigmoïd differentials membership functions.

Preferences are logical rules combining context concepts with infotainment description concepts.

```
Rule 1: IF Traffic is Jam
        AND (Destination Range is Near
        OR Destination Range is Local)
THEN Music Mood Fun is Important

Rule 2: IF Destination Frequency is Often
THEN Podcast Topic Politics is Important

Rule 3: IF Precipitation is Rain
THEN Music Genre Rock is Important
```

The first part of the rule is called the antecedent and the second part the consequent. Moreover, some context concepts can have a bigger influence on the filtering than some others; therefore we allow each rule to be given a weight.

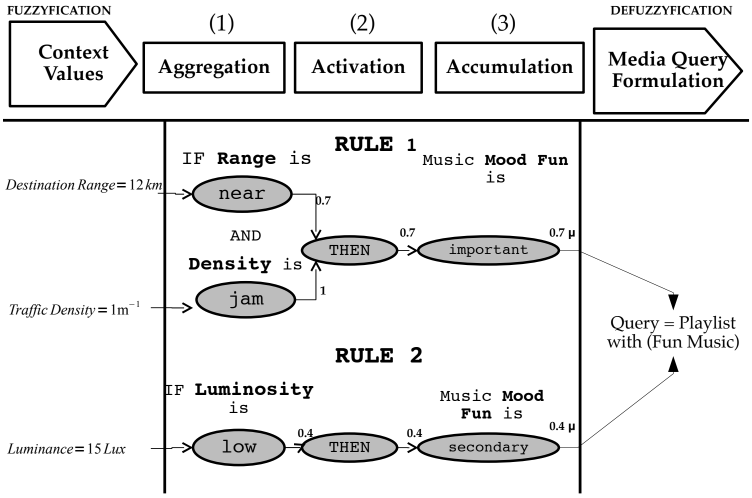


Figure 62: The inference of playlist creation queries through a fuzzy control process

#### 6.5.5.2 *Processing of the preference rules and inference of the playlist query*

Given a set of context values, we want to compute what would be the media choice of the user in this situation. We need to process each of the preference rules of the user profile in order to know the degree of truth of each of the consequents. This is called a fuzzy control process [, IEC] and runs in three steps (see figure 62): (1) during the aggregation process, sensors values are fuzzyfied and the membership functions compute the degree of satisfaction of the antecedents, (2) the activation is the computation of the logical implications of the rules, eventually (3) accumulation calculates the degree of truth of each of the consequents.

The aggregation consists in the computation of the degree of satisfaction of the logical condition of the rules, using $\mathrm{Min}$ (resp. $\mathrm{Max}$) Zadeh norms (resp. co-norms) for the conjunction (resp. disjunction) of the membership functions. For example, the rule `Rule1` defining a daily drive to work:

```
Antecedent 1: Traffic is Jam AND
(Range is Near OR Destination Range is Local)
```

is computed according as following:

$$a_1 = \mathrm{Min}\{\mu^{jam}_{Traf.}(x_{traf.}), \mathrm{Max}\{\mu^{near}_{Range}(x_{ra.}), \mu^{local}_{Range}(x_{ra.})\}\}$$

There are numerous fuzzy implication functions which can be used for the activation [Bouchon-Meunier, 2003]. The idea is to create a membership function defining the degree of satisfaction of the consequent. For the following, we use the minimum (Mandami's implication). In our example:

$$\mu^1_{Fun} = w_1 \cdot \mathrm{Min}\{\{a_1, \mu_{FunMusic}\}\}.$$

The value of the minimum is multiplied by the weight $w$ of the rule.

Every of the $n$ user rules $R_{k \in \{1,\ldots,n\}}$ generates an antecedent value, which is an evaluation of the degree of satisfaction of the condition of the rules. The role of the accumulation step is to make the mapping with the multimedia preference partition we have defined formerly, *i.e.*

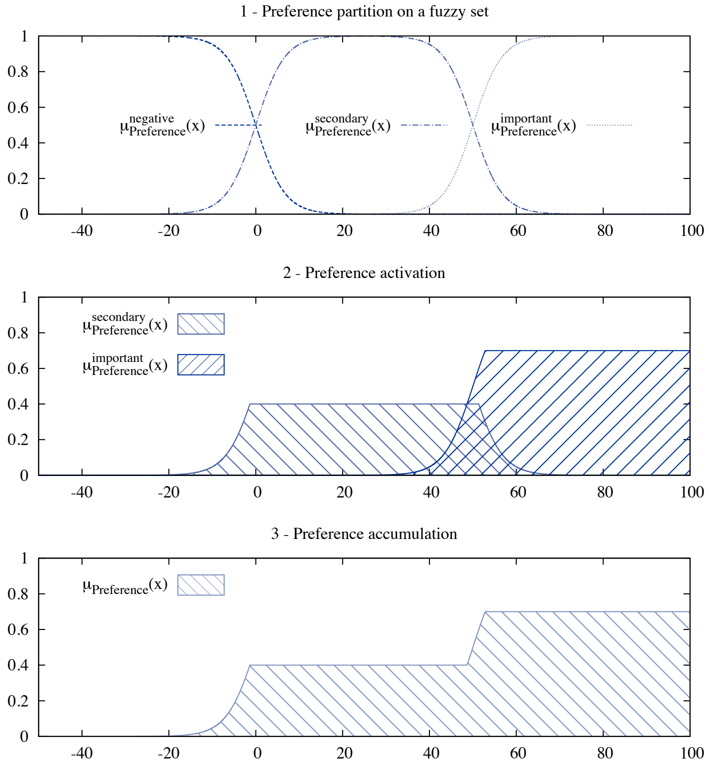


Figure 63: Accumulation of preferences for a given category definition of a media

to calculate the output for every consequent of the preferences on this partition. Consistently with the choice of a Mandami implication, we use a maximum operator as such $\mu_{\mathtt{Fun}} = \mathtt{Max}\{\mu^{k}_{\mathtt{Fun}}\}_{k\in\{1,\dots,n\}}$.

At this point, we have for each media description category a function $\mu_{\mathtt{category}(x_{\mathtt{category}})}$ with $x_{\mathtt{category}} \in [-50, 100]$ that expresses the degree of truth of this sort of implication:

`IF Context THEN Playlist Preference`

An example is given in figure 63, with two rules `R1` defining a condition where the music mood category is `Important` and `R2` defining a condition with music mood category `Secondary`. Given some context values the rules are activated with the values $a_1 = 0.7$ and $a_2 = 0.4$.

The last part of the inference process, is to determine for which value of $x_{\mathtt{category}}$ the degree of truth is maximum. Different methods have been proposed, we use the center of gravity of the function.

The inference process ends with the listing of the categories which formulate the channel creation query. After the aggregation step, each infotainment category has a value $x_{\mathtt{category}}$ defining the degree of activation depending on the context. We set a if $x_{\mathtt{category}} > 50$ then we add it to the query, otherwise we do not. The output of the reasoning engine is a query containing the categories of infotainment which are relevant according to the current context.

Listing 3: Algorithm for the activation and accumulation of fuzzy rules

```
For r in rules
 for a_r in antecedents of r
```

```
 set v_a context value of a
 fuzzyfy v_a
activate r
for c_r in consequents of r
 c_r x weight_R
for c in consequents
 c_final = Sum c_r with r in Rules
defuzify c_final
let query = {}
let c_min= 50
let c_pas=5
while {query = {}
 for c in consequents
  if c_final > c_min
  then query += c

 c_min-=c_pas
return query
```

### 6.5.6 *A local reasoning model based on tagged media content*

Since the complex situations depend on both on-board and off-board information data, the complex situation reasoner has to be a remote service. It enables context reasoning for the creation process of multimedia playlist, which occurs on the server side as we have detailed in chapter 4. However, in order to illustrate the advertising use case (*cf. suppra* 166) we have to implement some reasoning on the client. This is necessary because of the asynchronous character of the channel consumption with Personal Radio, *i.e.* once a channel is sent from the server.

LOCAL REASONING In order to achieve the advertising extension presented in chapter 4, the client is capable of deciding in a given advertising time-slot if the given advertising has to be displayed or not. For this we use pre-tagged advertising content with the following information: latitude, longitude and minimal distance. This information defines a circle on a map, if the vehicle enters in this circle, then the advertising can be played back in the next time-slot, other wise an alternative advertising (not tagged with content information is played back).

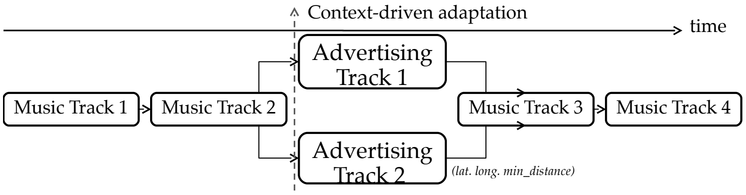


Figure 64: Context-based media switching for the advertising scenario

This use case illustrates how some reasoning based on simple situation can be achieved in a vehicle client using context-tagged content.

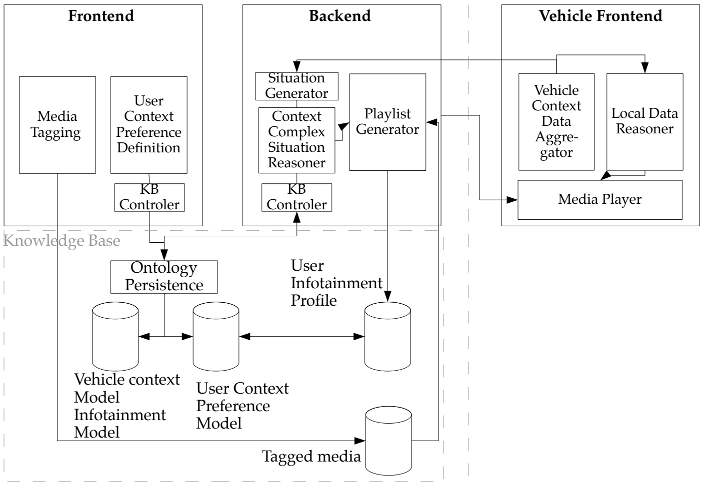


Figure 65: Functional architecture of the Program Director Framework

### 6.5.7 *Functional architecture*

The Program Director framework is based on a knowledge base that stores (1) the different models (context model, reasoning models) (2) the preferences of the users and (3) the context tagged media assets (advertising scenario). This knowledge base is coupled with a frontend which is used to populate it with context preferences defined by users, and context-tagged media (advertisement) defined by media producers (advertisers); and it is also coupled with a backend that processes context information in order to derivate complex situations and generates playlists which are sent to the mediaplayer of the vehicle (Personal Radio as defined in chapter 4). The functional architecture is depicted in figure 65.

#### 6.5.7.1 *The client*

The client reuses the functionalities of the client of Personal Radio as described in section 4.3.1. The local context reasoner is added, to the client so that it can process the relevance of an advertising using only the GPS position of the vehicle.

#### 6.5.7.2 *Server Frontend: Knowledge Base population*

The function of the frontend is to give to different sort of users the possibility to input to the knowledge base:

- User preference knowlege: Every user of personal radio can login to the frontend and can combine context concepts with infotainment concepts from the ontology in order to build rules. As we will see, only simple situations have been implemented.
- Media tagging: The advertisement generator which has be designed for media creator (like a marketing department) presented in 4.3.3 is extended, in order to be able to create geolocalised advertisements

#### 6.5.7.3 *Server Backend: Context Channel Business Logic*

The server backend contains the business logic for the creation of context channels.

CONTEXT COMPLEX SITUATION REASONER This reasoner implements the fuzzy control mechanism that we have described formerly. It is coupled with a situation generator which can derive from vehicle context information (for instance GPS position), other global context information (traffic conditions, weather, *etc.*). The Situation Reasoner can formulate a query to the playlist generator.

CONTEXT DEFINITIONS AND USER PREFERENCE MODEL PERSISTENCE This component is capable of loading the concepts and their logical dependencies from the ontologies and to transfert them to predicates in the fuzzy controler.

PLAYLIST GENERATOR The playlist generator is the same as described in previous chapters. It can aggregate different types of infotainment in a channel.

### 6.5.8 *Implementation of a proof of concept*

A proof-of-concept of the aforementioned components has been implemented and tested with the Personal Radio.

THE CLIENT The proof-of-concept reuses the client of Personal Radio and extends it with a basic context aggregator (using a simple interface input to give information like destination range, distance to next vehicle (for the traffic information, time of the day and GPS position) and a local context reasoner. The local reasoner, for the advertising use-case was embedded directly in the code of Personal Radio.

FRONTEND AND BACKEND SERVERS As for Personal Radio, the frontend server is a Liferay server and the backend a Glassfish server.The user frontend is a drag and drop web application. A user can select infotainment concepts, and drop context concepts on them to make simple associations. The figure 66 illustrates how this work.

The media producer frontend is the same as described in section 4.3.3 enables to give city as text-input and automatically find the GPS coordinates with the help of the Google javascript API. After that, the operator can select a distance-range defining a circle around the GPS coordinates. This circle is displayed on a map, in order to help the operator to figure-out, the coverage of the geo-localisation (see figure 67).

FUZZY REASONER FOR THE COMPLEX SITUATIONS The fuzzy reasoner is implemented using the jFuzzy library.[3] It is capable of loading the individuals concepts of the ontology and map them to antecedents and consequents in the rule bases of jFuzzy. jFuzzy allows different sorts of membership functions for the fuzzy sets, our implementation uses trapeze functions. In addition, a standalone client has been developed to be able to test the knowledge base. This standalone client is able to

---

[3] http://jfuzzylogic.sourceforge.net/

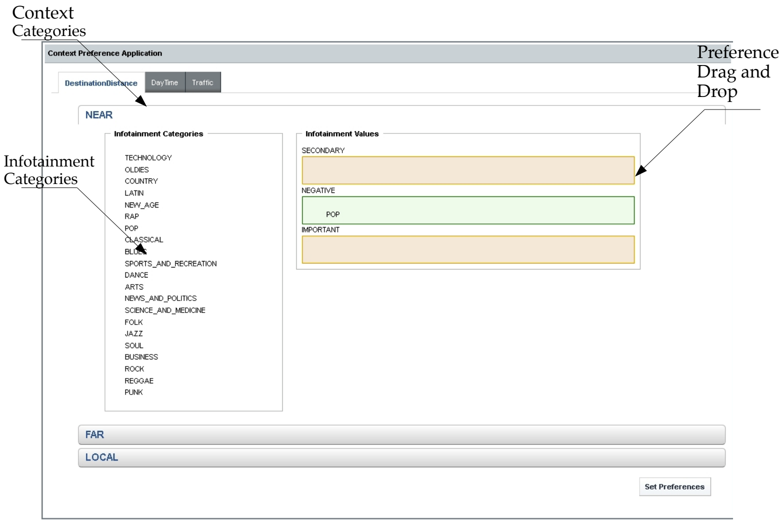


Figure 66: Definition of user preferences, using the drag and drop web application

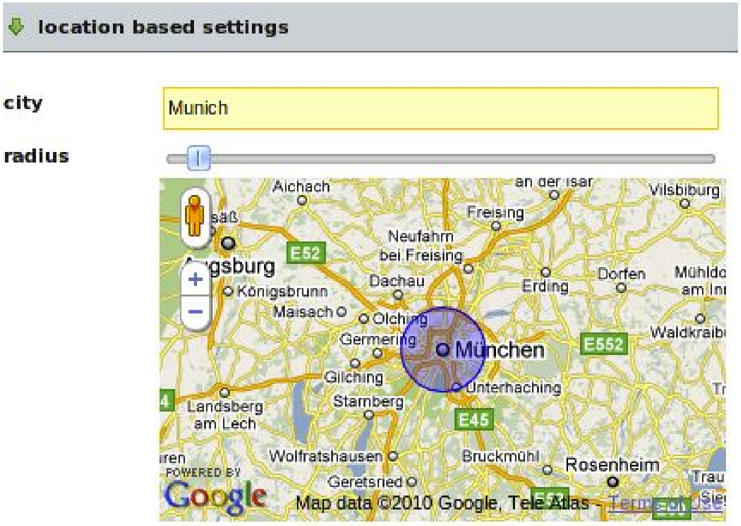
connect to the database load the ontologies and the preferences of a given user in the complex situation reasoner. It offers to a developer the possibility to test the situation reasoner, by offering a user interface to modify context-values. The results for all the infotainment categories of the knowledge base, are automatically computed and displayed (as on figures68 and 69).

**KNOWLEDGE PERSISTENCE COMPONENTS** The context of the user is not made persistent itself, since it could lead to a data privacy problem. Context data are used at computation time by the server in order to process reasoning, but are not saved anywhere.

- Ontology persistence: Ontologies are saved in a relational database (MySQL), using the SDB library of JENA[4]. SDB is capable to store different ontologie models in a same database. Jena SDB stores in different tables, the statements, the litterals and the resources [Wilkinson *et al.*, 2003]. Ontology models can be loaded in memory and manipulated as OWL graphs. All the classes of the description models: context model, and infotainment model derive from a fuzzy ontology, and the individuals of the classes define the fuzzy sets. Description models are saved in a same table to avoid loading the OWL graphs from separate databases, what is known to be rather unefficient. A separate database is used to save the graph which captures the preferences of the users.
- Tagged media persistence: Advertising tracks and their metadata information are stored in a separated MySQL database

---

[4] http://jena.sourceforge.net/



location based settings

city: Munich

radius

Figure 67: Tagging of advertising with localisation information

## 6.6 CONCLUSION

The need to reduce the cognitive load of secondary tasks in vehicles leads to the design of expert-driven systems in various fields. The infotainment systems which are getting always more complex in the possibility that they offer to the user, offer a wide range of passive personalization possibilities. The Program Director Framework is an answer to a global approach of context sensitivity in vehicle embedded infotainment system:

- A context model which captures the different context information of a vehicle and enables persistency of the context adaptation policy
- A reasoning method based on a fuzzy controller which takes into account the semantization of raw sensoric data
- A reasoning architecture adapted to the requirements of vehicle infotainment

Our work has pointed out the necessity to couple reasoning with effective knowledge persistence of the adaptation policy. Moreover, we have demonstrated that local and remote decisions in terms of context-reasoning for mobile environment are both necessary but have to rely on different techniques based: (1) on the amount of context available and (2) the computing capabilities. An isolated client which has a partial knowledge of its context and of the available resources, can only make some partial change to the rendering to finalize the context adaptation whereas a central server, with an exhaustive knowledge the situation and of the available resource can pre-filter much information.

Figure 68: User interface for the situation reasoning simulation: Context value input

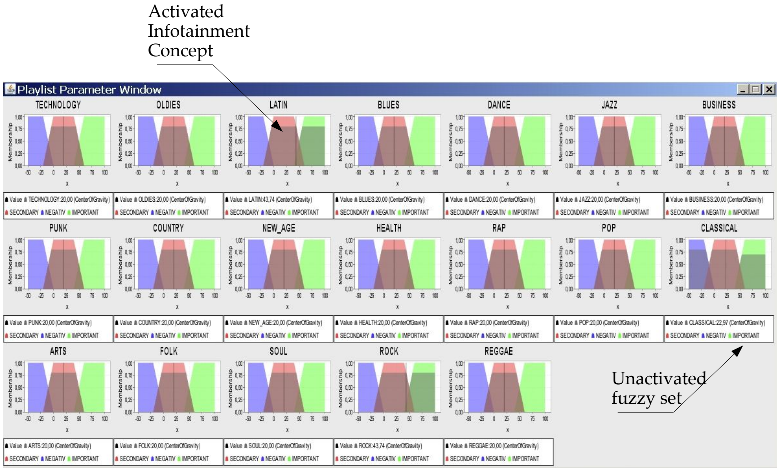


Figure 69: User interface for the situation reasoning simulation: Infotainment value output

# 7 EVALUATION OF THE CONCEPTS

> *" The only difference is that since the Street does not really exist – it's just a computer-graphics protocol written down on a piece of paper somewhere – none of this things is being physically built. They are, rather, pieces of software, made available to the public over the worldwide fiber-optics network. When Hiro goes into the Metaverse and looks down the Street and sees buildings and electric signs stretching off in the darkness, disappearing over the curve of the globe, he is actually staring at the graphic representations – the user interfaces – of a myriad different pieces of software that have been engineered by major corporations."*
> — Neal Stephenson, *Snow Crash, 1992*

The concepts which have been presented in chapters 4, 5 and 6 cover a wide range of topics in the multimedia field, the telecommunication field and computer science field. While a global study of all possible validation approaches of vehicle multimedia applications goes far beyond the scope of a single thesis, we are going to assess the different concepts and prototypes that have been developed so far, with regard to the literature about vehicle specific and non-specific multimedia evaluation. Based on this study we can combine expert and experimental methods to draw an analysis of the work which has been achieved.

## Contents

Since loads of software and hardware components have been specified and most of the time developed for this thesis, it .

### 7.1 EVALUATION OF THE ACTIVE PERSONALISATION

The user active personalisation that we have introduced in chapter 5 have two goals which have to be evaluated:

1. the mitigation of the user distraction
2. the mitigation of the user frustration

The conception of the prototype has been , and the second one by the integration of a recommendation technique in an architecture which scales to the user information need and which accounts for the known limitations of computing and networking resources of the headunit of a vehicle.

### 7.1.1 *Literature review : Evaluations of user interfaces for vehicle systems*

A systematic study of IVI evaluations techniques has been lead by Young and Regan [2007], by Pettitt [2008] in his PhD Thesis and Burmester *et al.* [2008]. More specific vehicle multimedia applications, like the use of a search engine or the browsing of a list have been assessed by Jeon *et al.* [2009] and by Graf *et al.* [2008]. The differentiation between the techniques can be drawn according to two dimensions: (1) the method can be experimental (most of the time with a user panel) or based on expert knowledge and, (2) the object of the study which can be quantified; They can be ergonomic measurements (time for task completion, task completion success rate, primary task deviation) or a qualitative user acceptance (like the subjective pleasure to use the interface or an assessment of the advantages and drawbacks of the application from a user point of view).

#### 7.1.1.1 *Task performance evaluation*

Since user interfaces are intended to allow the user to process a task, researchers have tried to develop systematic methods to assess the performance of the task completion.

**THE 15 SECONDS RULE** The 15 seconds rule that we have introduced states (*cf. infra p.*127) that no task should last longer than 15 seconds for its completion. Even if this rule is widely used for its simplicity, it is rather difficult to apply to the criteria selection of our music browser. Indeed, the user must select at least one criterion and can define up to five or six criteria. The more criteria are selected, the longer lasts the task completion. What is important is "the ability of drivers to complete the task in chunks of interactions" [Pettitt, 2008].

**VISUAL MODALITY EVALUATION TECHNIQUES: OCCLUSION TECHNIQUES AND GLANCE MONITORING** This reference to so called "chunks of actions" are based on the second and the third rule that we took into consideration regarding the duration of the user glances. Occlusion techniques rely on the assumption that by forcing short user glances to the interface, it is possible to recreate the visual load of the driving behavior. The vision of the user is artificially altered using shutter or goggles which alternatively hide and reveal the visual scene [Young and Regan, 2007, Graf *et al.*, 2008, Rölle, 2007]. The ability to perform the task can be evaluated (time to completion), or the capacity to interrupt and resume the task (resumability ratio) under different glance durations and frequencies. This rule has been standardised by a sequence of standards. The validity of the technique has been extensively discussed, and is quite heavily criticised by Pettitt [2008] for its lack of sensitivity, the resumability being the only interesting and reliable information that can be extracted. Another way to evaluate the visual modality is to monitor user glances with the so-called eye-tracking systems. Thereby, it is possible to assess how often the user changes between visual information such as like looking at the road and looking at the infotainment display [Burmester *et al.*, 2008]. The technique has been standardised as well. Both techniques are quite time consumming and only deliver information on the distraction aspect.

#### 7.1.1.2 *Resource competition tests*

Infotainment application belongs to the third priority-class of the driving tasks (*cf. infra p.* 126) and the influence of resource competition can be measured in order to evaluate the effects of the user distraction on his driving behavior.

LANE CHANGE DETECTION Unlike the former method, which involves a non driving specific primary task, the lane change detection method has been widely developed to assess both the visual and the cognitive load [Lansdown, 2000]. Whilst performing the task on a prototype of the infotainment system, the user has to perform a driving task by following directions on a simulator which creates a road environment with more or less realistic traffic conditions. The position taken by the user on the road, the distance to other vehicles and the reaction time to direction changes are recorded and compared to a normative reference. The lane change detection (LCT) has been normalised.

ON ROAD STUDIES AND DRIVING SIMULATOR On road studies are certainly the most ideal way to assess a user interface since they are the closest to the usage conditions of the final application. Young and Regan [2007] has listed the measurements that can be recorded whilst the testdriver performs concurently his driving task and the application test: visual scanning (through eye-tracking), task completion, tester subjective workload assessment, reaction time, distance to preceding vehicle, *etc.*. Simulator, can recreate conditions which are very close to reality and are helpful when testing application which could have major safety applications.

VIGILANCE TASK SIMULATION When no driving simulator is available, and on-road studies are too expensive, it is possible to force users to test the application as a peripheral task (*e.g.* the browsing of a list) by performing a primary task also called vigilance task (*e.g.* a game where the user has to catch balls falling on a screen) like in the study of Jeon *et al.* [2009]. The time for the task completion as well as the primary task completion success (*e.g.* ball catching rate) can be recorded to assess the user distraction. Pettitt points out that even if it is alleged to give valuable results, this method is not well standardised, and therefore the result of such studies are difficult to compare [Pettitt, 2008].

#### 7.1.1.3 *Heuristic and qualitative evaluations*

Quantitative user studies need large panels and optimal test conditions which make them difficult to perform systematically when developing a new interface. Moreover, they give little information on a crucial element of infotainment system: the subjective impression of the user.

It is thinkable that pure quantitative tests on ergonomics cannot deem a fairly unpleasant application, since its ergonomic design has been well thought-out but its final output (*e.g.* music recommendation) is not relevant to the user or has a disapointing quality. For all of those reasons, application designers and psychologues have developed methods using heuristic models of the cognitive load or qualitative user panels.

COGNITIVE MODELS We already mentionned in section 6.3.1 that the understanding of the mental capabilities of an individuals in computer science research and especially in HCI research has been largely influenced by the activity theory. The most highly cited model has been developed by Card *et al.* [1983], and is called the GOMS for Goals, Operators, Methods and Selection Rules. It consists in splitting every user task in different goals which can be achieved within a list of elementary actions which are called operators. As there can be different types of operator sequences, they can be grouped in methods, and methods themselve can be combined differently according to selection rules.

Even if GOMS represent the advantage to avoid quantitative user studies, but are still complicated to implement for researchers, especially in a fast-prototyping development process 2.9.3.1. The Key-Stroke Level models (KLM), which are a simplification variant of GOMS have been recently discussed as possible prediction technique of occlusion metrics, *i.e* using a KLM method, it could be possible to predict the results of the occlusion metrics. Even if the results are very promising (see [Pettitt, 2008]), the validity of such models is still quite subject to discussion. That does not imply that heuristics deriving from cognitive models should be crossed-out from any evaluation procedure of a vehicle user interface, but their role in the whole development process should be better understood in order to avoid wasting resources form unreliable information.

QUALITATIVE EVALUATION Qualitative evaluations cover the domain of sociological study around infotainment applications. They intend to evaluate the applications within the context of the user. Context meaning, their personality traits, technical interests and the intended usage of the application. Ethnographic approach have been developped by sociologists who wanted to measure the user or customer acceptance for new types of products. The Attrakdiff model has been developed by Hassenzahl [2004], Burmester *et al.* [2008] [1] to evaluate the user stimulation potential of a product. A set of testers is asked to answer a questionnaire of so called semantic differentials which are lists of bipolar adjectives (for example: "Adequate-Inadequate", "Good-Evil" or "Valuable-Worthless" [Wikipedia, 2010b])[2]. In a pure multimedia domain, Pauws and Eggen [2002], Pauws and van de Wijdeven [2005] combined a user qualitative evaluation with an evaluation of the task performance to evaluate an interactive playlist generation prototype called Satisfly. He underlined that recommender systems in the music domain should decrease the time needed and the number of tasks when creating a playlist. Vignoli [2004] organises the development of a new

---

[1] A complete review of the bibliography about Attrakdiff is available on their website: http://www.attrakdiff.de/AttrakDiff/Publikationen/

[2] The semantic differential rating scale and the notion of bipolar adjectives have been developed by the psychologist Osgood [1975]

music playlist creation prototype in a two-step process: (1) First he makes semi-structured interviews on a relatively small set of users to identify what are the key expectations for the prototype to be designed, (2) he then makes a review of the existing products and (3) ends with user experiment of the prototype. To give a proper directions to the step of (1) and (3), Vignoli made a preliminary Web Study in order to get quantitative information guiding the step (2).

### 7.1.2 *Summary of evaluation methods*

Evaluating a new multimedia system and specifically its user interface, does not only belong to the end of its development. It should be actually an on-going taks of any application design project. For this reason looking for new use-cases and new scenarios without a specific knowledge of user's expectations makes little sense. The best user studies combinate qualitative and quantitative in order to rely on the advantage of both techniques since they give different sort of information on a application (see table 18). Quantitative information are very helpfull when trying to compare solutions to specific problems under reproducible conditions but give little insight on the subjective evaluation of a user. Statistical correlation does not mean logical causality, and therefore, qualitative techniques have the advantage to give more concrete information on the user expectations.

In the automotive domain, when evaluating the distraction potential of an application, quantitative user evaluations are the most valid but they are also the most expensive. Expert methods, have been generally standardised in HCI but their reliability still fall for short of expectations. Actually increasing research efforts have lead to the standardization of ergonomic conception methods, in order to minimise the testing overhead in the development process and to predict the user acceptance. In his doctoral dissertation Hummel [2008] has established that the personality traits can play a certain role when leading acceptance tests.

In the design of our prototype a strong focus has been given to the implementation of an efficient architecture which achieves optimal reactivity performances and thereby lessen the user distraction. Indeed, we can evaluate the choice that we have made according to the requirements that have been already defined with an acceptable accuracy (see section 5.2). Because of this cost/accuracy ratio, we have decided not to evaluate experimentaly the user distraction but rather the user acceptance in an on-road study.

### 7.1.3 *Expert qualitative evaluation*

The design of the application has been discussed with experts in the multimedia and HCI fields at different occasions (intern discussion at BMW, discussion with development partners from the Institute of Artificial intelligence at the University of Ulm, discussions with other researchers at multimedia workshops in Florence[3], and Klagenfurt[4].

---

[3] ACM International Workshop on Mobile Cloud Media Computing

[4] 6th Symposium of the WG HCI&UE of the Austrian Computer Society November

|  | Meth. Type | Assessment Domain |  | Cost | Standard. | Validity |
|---|---|---|---|---|---|---|
|  |  | Driv. task | App. task | Pettitt [2008] |  | Pettitt [2008] |
| Occlusion test |  |  | ✔ | ☆★★ | ☆★★ | ☆☆★ |
|  | User | Task completion performance and resumability |  |  |  |  |
| Lane Change Detection |  | ✔ |  | ☆☆★ | ☆☆★ | ☆★★ |
|  | User | Deviation, reaction time |  |  |  |  |
| On-road studies |  | ✔ |  | ☆☆☆ | ☆☆☆ | ★★★ |
|  | User | vis. scanning, reaction time, subjective eval. |  |  |  |  |
| Vigilance Task |  | ✔ | ✔ | ☆★★ | ☆☆★ | ☆★★ |
|  | User | vis. scanning, reaction time, subjective eval. |  |  |  |  |
| Cognitive Models |  | ✔ |  | ★★★ | ☆☆★ | ☆★★ |
|  | Expert | Heuristics |  |  |  |  |
| Qualitative Study |  |  | ✔ | ☆★★ | ☆★★ | ☆★★ |
|  | User | Subjective quality, improvement suggestions |  |  |  |  |

Table 18: Human machine interaction evaluation techniques for the automotive domain

| | |
|---|---|
| Multimedia aspects | Creating the playlist based on seed songs does not guarantee a sufficiant access to the whole catalogue. It could be necessary to evaluate the quality of the caching in order to be able to assure that sufficiant data has been cached |
| | Externalising the playlist generation to third-party providers (such as the Echonest) is an interesting technique, since it makes a separation of concerns between content provider, recommender service and device. |
| HCI aspects | The whole time needed to create a playlist is depends on the number of parameters that users want to use and therefore this is going to influence the obtrusivity. |
| | The fast preview mechanism does not give the impression that the content displayed is just a sample of the whole content. It is a very interesting aspect of the application to have a preview while formulating the query. |
| Suggestions | The use of preference relaxation in the formulation of the query can be a mixed blessing. On the one hand, it makes sense to give more weight to the first criteria which are selected by the user than to the last ones, but on the other hand, it depends in which order they are selected. |
| | The caching could be implemented localy on the client (as we suggested in [Hahn *et al.*, 2010]) to make the application faster. |

Table 19: Summary of the discussions

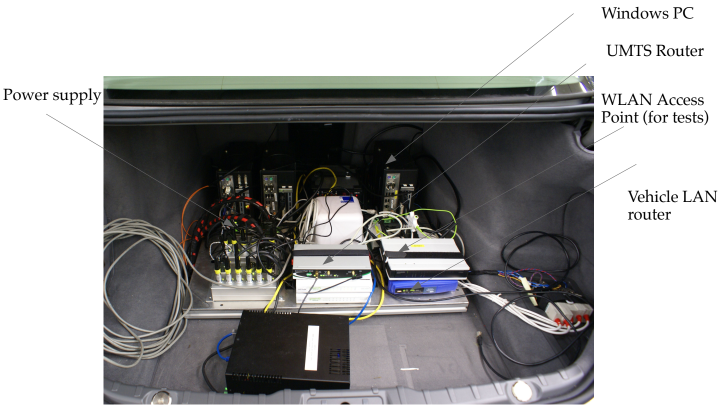


Figure 70: Integration of the vehicle hardware in the coffer

### 7.1.4 *A on-road qualitative evaluation of the prototype*

We have decided to make an on-road evaluation of the prototype using based on Attrakdiff questionnaires and semi-conducted interviews. The study itself was conducted in german by Sascha Gebhardt and is largely described in Gebhardt [2010]. We reproduce in Appendix the questionnaires which had to be answered by the participants.

#### 7.1.4.1 *The evaluation set-up*

The study took place in a BMW 7 series that had been modified in order to be able to run the application that we have described in chapter 5 (see figure 70 for the integration of the hardware elements in the coffer.

#### 7.1.4.2 *The evaluation procedure*

The evaluation was split in two parts:

1. First a parking phase. The vehicle standing in park position, the participants sitting at the driver's place were asked to familiarise with the application menus on their own. After a short period of time (five to ten minutes) they were explained how to use the application in details and had to fill in a first AttrakDiff questionnaire.
2. Afterwards the participant had to drive on a given track in the northern outskirts of Munich (see map on figure 71) mixing both urban with semi-urban roads and superhighways. During their drive, they had to complete different tasks using the application and three of them were evaluated with semantic qualifiers: (1) set filters and start a playlist, (2) play a song from the history and (3) play a song from the album of the current song. A the end of this driving phase, a second AttrakDiff questionnaire was filled in by the participants regarding their whole impression of the application.

During the whole evaluation, the participants were asked to think loudly so that notes on their impressions could be recorded.

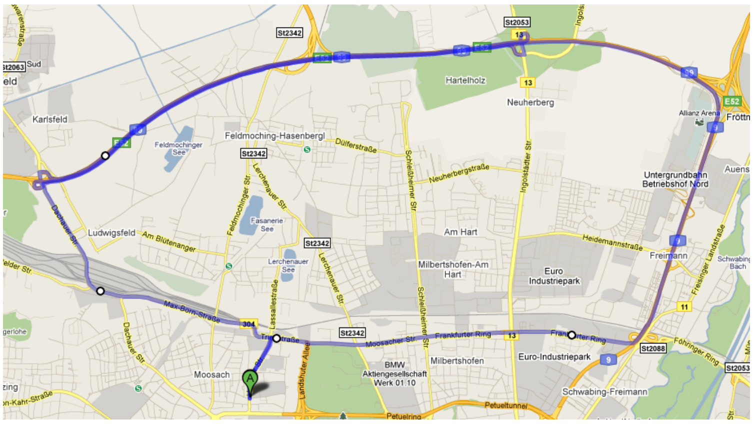


Figure 71: The road track which was taken by the participants.

| | |
|---|---|
| Positive | Most of the participant found the application pleasant to use. One even asked if such an application could be available for a desktop use. |
| Negative | The preview function lead to some misunderstandings. Some participants tried to select items within the preview even if this functionality was not turned on. The removal of filter criteria in the formulation of the query was not well understood. |
| Suggestions | Pause function. One participant suggested that the possibility to create playlist should be turned-off when driving. Some participants took advantage of the stops at firelights (between 10 to 40 seconds) in order play around with the application. |

Table 20: Results of the semi-conducted interviews within the on-road study

#### 7.1.4.3 *Results of the evaluation*

Each single evaluation lasted roughly fourty-five minutes and we were able to schedule 18 participants (16 males and 2 females with an average age of 30.6 years old) within less than a month period when the vehicle was available. Unfortunately, the UMTS router that we used was defective, and even if we managed to change it before the end of the study only 11 participants could complete the whole study (10 males and 1 female). Our participants were for security reasons all in possession of a BMW specific driving licence and therefore; they were all already familiar with the iDrive.

THE QUALITATIVE EVALUATION A lots of impressions that have been recorded concern pure design aspects of the application and we will not analyse them here since it is not the focus of this thesis. Regarding the functional aspects we can summarise the remarks of our participant as in table 20.

ATTRAKDIFF Based on the semantic differentials which consists in evaluating the sequences of bipolar adjectives we could compute different qualitative evaluations of the application and of its usage. Globally we observed an improvement of the hedonistic and pragmatic judgement after the driving phase (see figure 72). This can be explained by

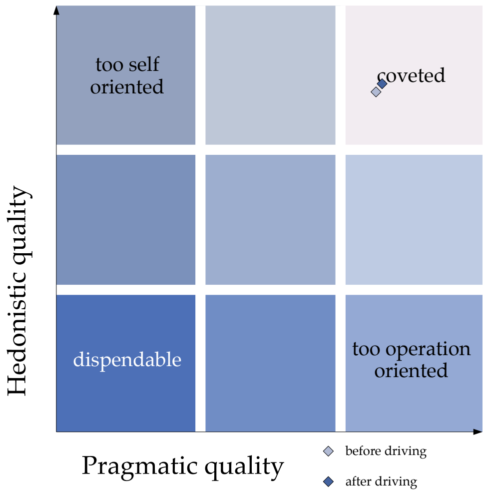


Figure 72: AttrakDiff matrix before and after driving

the fact that participants get familiar with the application by using it and therefore find it more pleasant and more convenient to use, but also because the application is better suited for driving since it has been specifically designed for this.

The evaluation of the tasks was largely positive but we noticed that the activity was always the worst evaluated among the characteristic of the application. This is to be compared to the comments that we could record during the on-board study. Even if we have simplified the tasks, the way to process them was not always completely understood and the learning curve remains a parameter which is difficult to monitor in a 45-minute test. We can explain this by the discripencies between the vehicle ergonomics and a desktop ergonomics which demand a certain adaptation of the user in the short run but which tend to give better result after a long training period. This assumption is to be compared with the study of Rölle [2007], who evaluated the learning curve of his vehicle music navigation interface: he noticed that the users need some time to adapt to the vehicle specific interface but that in the long-run, the success ratio of the vehicle specific interface is better.

### 7.1.5 *Stress evaluation of the architecture for the heuristic cognitive evaluation*

We have also evaluated the capacity of the architecture to give a solution to the service latency among the different service providers that we have mentioned in chapter 3. We have decided to focus on the time necessary for the client to retrieve a playlist once filters have been selected by the user. More precisely, we have focused on the reactivity of the server to the request of the client since, over types of processing (client request formating, server answer processing) have a minor influence on the overall performance of the web-service application [Chekir, 2007].

To perform this evaluation we have simulated the playlist preview and playlist requests which were sent to the server using the curl[5] software. Curl can measure a lot of statistics over the HTTP protocol (time to initiate a TCP connection with the host, time to start transfer,

---

[5] http://curl.haxx.se/

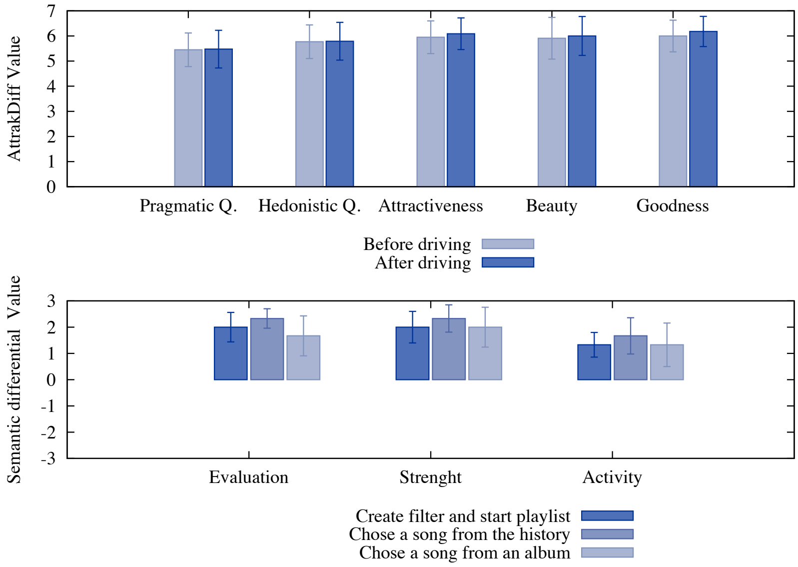


Figure 73: Attrakdiff evaluation of the application and semantic evaluation of the tasks

time to full download, *etc.*). Curl can also simulate different throughtput bitrate so that we could perform the tests among both other a UMTS connection or other a local connection in order to save data traffic plans.

#### 7.1.5.1 *Preview time and playlist generation is not dependant to the number of criteria*

First, we simulated a single client processing different preview and playlist requests with varying number of criteria. Without surprise we have noticed a constant time for those measures, independantly of the number of criteria (see figure 74). This is due to the structure of our server database where the different cached criteria are saved as foreign-keys in the track table. Therefore, the computation of the preview is a simple `SELECT` operation with different conditions depending on the structure of the request.

#### 7.1.5.2 *Evolution of the preview time depending on the number clients*

We have also simulated the charge of the architecture when using a number of clients between 5 and 20. A script running parallel clients arriving to the server with a Poisson's law of parameter $\lambda$ could give us an idea of the capacity of the architecture to assure reactive service under stress. We did two simulations, in the first simulation, the server has a pool of 10 threads, which means that he can concurrently request 10 service providers, if further connections to third-party providers are requested, they are queued by the internal scheduler. In the second simuluation the server has a pool of 25 threads.

Unfortunately, we encountered many problems with the internet connection that we could use to perform the tests. It turned out that the capacity of the symetric digital suscriber line that we were sharing with other researchers was largely under provisioned and therefore, at some moment completely overloaded. For this reason we decided to run the on-road studies (which implied a single client) with a privately hosted

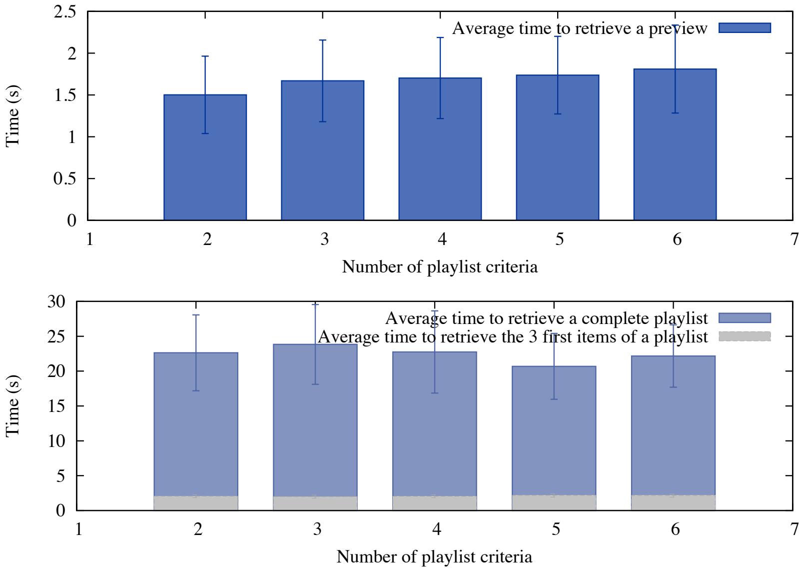


Figure 74: Time to display a preview and to create a playlist according to the number of criteria

server (which was connected to the internet with an asymetric line and therefore, unable to properly run many concurrent service requests).

Even if the results of the study must be precautiosly interpreted because of the difficulties to simulate a stable charge due to a highly varying network usage of our research platform, some tendencies in the figure 75 can be extrapolated.

The intensity of the stress (the arrival frequency of the clients) has a bigger influence than the charge of the stress (the number of client requesting) on the computation of the whole playlist than on the preview time. This can be explained as following: Thanks to cached data, the preview of the playlist can be served without requesting concurently different providers. The creation of the playlist is much more impacted by the intensity of the stress since, the server needs to concurently access different service providers. This tests show that even with very disadvantageous network conditions, the computation of the preview remains under acceptable levels, especially when the server is granted enough threads (the average preview time remains notably under 2 seconds with 20 clients, if the server is granted the maximum number of threads).

### 7.1.6 *Quality of the cached data*

The caching of the data has a drawback, it reduces the amount of tracks that the user can discover with the filter/preview process in comparison with a direct access to the whole catalogue. When creating a playlist only the cached tracks are used. A suggestion of the expert review of the prototype was to analyse the caching and especially its representativeness of the whole catalogue. We have measured the quality of the caching with the two metadata provided by the echonest: (1) the hotttness and (2) the familiarity. The first being a sort of time derivate of the second (*i.e.* the more an artist get known, the 'hottter' he is). Using the API of the Echonest we have measured the quality of the

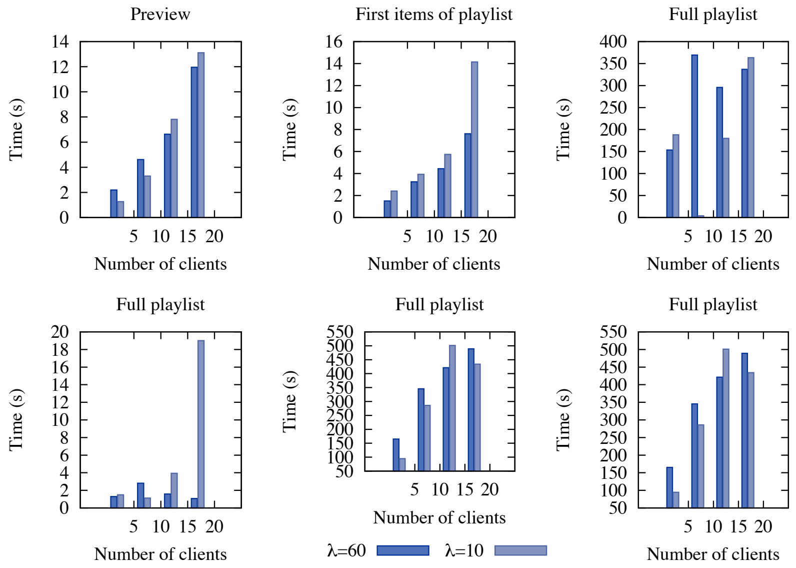


Figure 75: Stress tests of the architecture with different charges of clients and different number of concurrent threads for the server

cached data that we gathered from Gracenote (data containing mood information) and from Rhapsody (data containing genre information).

The hits (see section 5.1) are clearly identifiable in the right parts of the histogram. They are logically few but still well represented. The long tail is composed of tracks of an average familiarity (resp. hotttness), some tracks have a very low familiarity and we can assume that under a certain level of occurence, the echo nest gives normalised values this is the reason why the curve presents local peaks for very low values. This repartition of the tracks assure us of a balance between current hits and mainstream artists. It is also thinkable to monitor regularly the quality of the cache, in order to assure that the cached data makes a representative sample of the remote catalogues.

## 7.2 QUALITATIVE EVALUATION OF THE CONTEXT ARCHITECTURE AND DISCUSSION

Experimentally evaluating a context-driven system can be a very cumbersome, especially when no ground-truth data is available to guide the assessment of the adaptation. For automotive scenarios, context-adaptation evaluation has been limited to driver assistance scenarios under hard environment constraints that make them difficult to generalise. Kazi-Aoul [2008] evaluated the charge that the distributed context adaptation that she describes, represents on existing web architectures. Kaltz [2006] evaluates the Catewalk prototype by describing a case study followed up by a scientific discussion. General studies of mobile context scenarios lead mainly to the evaluation of a list of requirements[6].

---

[6] Weiß [2009] for example lists fourteen criteria for the evaluation of a context-driven architecture: support of different mediatypes, formal description of the content, formal description of the user, formal description of situation description rules, enhancement easily possible, support of different personalisation techniques, support of user-generated content, implicit profiling, techniques to measure similarity, application independancy,

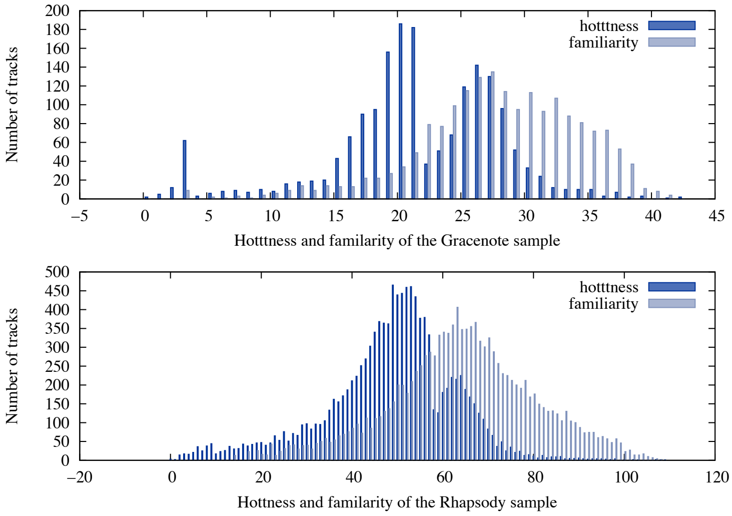


Figure 76: Distribution of the cached data in terms of Hotttness and Familiarity

### 7.2.1 *Genericity of the context adaptation*

The genericity of the context adaptation can be measured mainly on three dimensions:

CAN THE ADAPTATION BE PROCESSED ACCORDING TO ANY KIND OF CONTEXT INFORMATION? The fuzzy controller that we have presented in section 6.5.5 enables reasoning on a very wide range of context values, since it is capable to map from raw sensoric data to semantic rules. However, concepts which cannot be described as fuzzy sets (like the driving intention) have to be approximated by fuzzy predicates.

CAN THE ADAPTATION BE PROCESSED ON ANY SORT OF MULTIMEDIA CONTENT? Metadata describing the media assets are *de facto* generic since they are designed to systematicaly described, classify and sort media content with textual information. The more high-level are the metadata (*i.e.* the closer they are to the human 'common sense'), the more generic is the rule defining the adaptation. The framework enables the use of metadata describing audio or video information, and all kind of semantic description of the content (*e.g.* music genre, podcast topic). Therefore, the framework that we have presented is not dependent to a specific type of content.

IS THE TYPE OF ADAPTATION DEPENDENT OF THE FUNCTIONALITY WHICH HAS TO BE ADAPTED? The type of adaptation which is processed is not indepedent from the functionality. The localisation of content (in the advertising scenario) is processed remotely, but the decision when to playback is taken in the vehicle. Whereas for the creation of context playlists, the aggregation of adapted content is processed remotely based on vehicle informations.

---

protection of the privacy of the user, limite usage of the mobiles network interfaces, scalability, efficient processing of the personalisation.

### 7.2.2 *Optimisation of the vehicle ressources which are available*

DO THE SOFTWARE SCALE TO THE COMPUTING RESSOURCES NEEDED FOR THE CONTEXT ADAPTATION Depending on the type of adaptation (geolocalised advertising) or (context-based playlist) (1) the amount of information to make the decision and (2) the amount of processing resources for the context reasoning are fairly different. We have observed that querying semantic graphs is quite resource consuming this is the reason why we have decided to base the program director framework on a proxy architecture where most of the adaptation is processed remotely. However, some types of decision, like the display of an advertisement in a defined region need very few data information and therefore we have decided to implement them in the vehicle.

### 7.2.3 *Functionality aspects*

WHO DEFINES THE ADAPTATION RULES? The rules for the adaptation can be defined with a web user interface or using an OWL editor[7]. The drag and drop of user preferences can limit the complexity of such a task even if the number of possible combinations between playlist descriptions and context descriptions is combinatory exponential. Complex conditions, like the combination of different predicates is not possible yet with the web interface.

DO THE USER STILL HAVE CONTROL OF THE MULTIMEDIA APPLICATION? When a context-based playlist is generated with this proxy architecture, it is still a playlist and the user have the same control on it as with any other type of playlist. The introduction of context-driven features raises actually the question of application transparency.

## 7.3 CONCLUSION

The evaluation of a new multimedia applications is always a trade-off between the genericity of standardised evaluation methods and the specificity of the goal to achieve. The experimental results of the user interface evaluation have shown a strong user acceptance of a highly configurable music browsing interface for a vehicle scenario. The architecture that we have designed for the passive personalisation combines the advantages of a proxy architecture where the decisions are taken remotelly when possible, so that the client adaptation software remains simple. However, for some types of adaptation, like location based advertising, the adaptation decisions are taken locally in the vehicle.

---

[7] The ontology for the persistance of user rules can be edited with Protégé, an ontology editor developed by the Stanford Center for Biomedical Informatics Research: http://protege.stanford.edu.

# 8

# CONCLUSION AND FUTURE WORKS

*« Nous sommes comme des nains juchés sur des épaules de géants (les Anciens), de telle sorte que nous puissions voir plus de choses et de plus éloignées que n'en voyaient ces derniers. Et cela, non point parce que notre vue serait puissante ou notre taille avantageuse, mais parce que nous sommes portés et exhaussés par la haute stature des géants. »*
— Bernard de Chartres

## Contents

This thesis deals with the integration of on-demand media services in the vehicle. Internet services have to be adapted before being stagged into vehicle software platforms. Indeed, the vehicle infotainment software is part of manyfold embedded systems with different requirements. We have focused on the following requirements such as the complexity of the software, the computation resources and the network access resources. Moreover, making on-demand services available in a vehicle has functional requirements in terms of ergonomics and user obtrusiveness.

## 8.1 CONTRIBUTIONS OF THE THESIS

The chapter 2 establishes that the vehicle integration of new media consumption paradigms raises both functional and technical challenges: (1) the need for the diversity of on-demand entertainment solutions augments with the growth of individual transportation time budget and (2) the vehicle already offers a rich set of hardware devices for the network access, the media control, and display which can be programmatically accessed for the development of new functionalities. The metadata which have appeared together with digital media, offer a wide range of personalisation potential. The chapter 3 explains how the metadata can be used for media search and recommendation and position the aggregation of metadata from multiple source as a new dilemma to be solved.

The different prototyped solutions presented throughout this thesis contribute to give concrete solutions to our problem statement.

### 8.1.1 *Contribution to the multimedia delivery infrastructures for online content*

In chapter 4, we have presented the personal radio platform as a solution to on-demand content. The personal radio is based (1) on a rich client which supports different audio and video codec standards, and which have been integrated to current automotive HMI, and (2) on a layered web-architecture including:

- A backend with an abstraction layer of content providers which helps minimizing the client complexity. The client can access common catalogue search functionalities of the different providers and download directly content.
- A frontend which provides administration for the creation of personalised channels. We have illustrated how the on-demand architecture could be extended with a customer relationship management system to provide selected information about products.

### 8.1.2 *Contribution to the design of reactive application in the automotive domain*

We have presented, in chapter 5 a prototype of a music browser for the automotive use. The user interface and the architecture of this online music browser have been designed around vehicle haptic and video interface modalities to minimize user distraction and avoid user frustration. With this browser, the user can with a few interactions create an highly personalised playlist and start instantly consumming it. It involves a caching/aggregation technique to use and combine metadata coming from different providers. We have established that the quality of the caching can be measured.

### 8.1.3 *Contribution to the architecture of context-sensitive applications for the infotainment domain*

Finally, the passive personalisation turns out to be a solution to reduce the cognitive load of media search and selection. The chapter 6 describes a context model adapted to both local and remote decisions. It involves a reasoner based on fuzzy logic capable of mapping values from the sensoric to logical user preference statements. The reasoner is integrated within personal radio as a combination of a proxy adaptation, which can be directly integrated with the personalised playlist creation components presented in the previous chapters.

## 8.2 FUTURE WORKS

Through the different chapters we have highlighted potential future research directions.

### 8.2.1 *Application user transparency*

The transparency of the application is a key issue which has to be solved with the boom of vehicle functionalities especially in the infotainment domain. This transparency covers mainly two aspects: the user interface and the user privacy.

#### 8.2.1.1 *Contexual user interface: User dynamic awareness*

Depending on the context (*e.g.* vehicle standing or driving, network connection available/limited/not available), functionalities might not be available or be degraded or even be banned from using (like TV while driving). For instance the creation of a personalised playlist when no connection is available should still work but only with local content. There is a challenge in informing dynamically and transparently the user with such changes, without obtrusiveness and frustration.

#### 8.2.1.2 *User privacy: preference policy dissemination*

In our vehicle/remote-proxy architecture, user preferences are stored online and context information is processed both in the vehicle and on a remote service. This service, is in our scenario assured and managed by the car manufacturer. Of-course, more interactions of the user preference with the home environment or the mobile environment would be possible in a scenario of superdistribution of content between multiple devices. However, the dissemination of infotainment preferences and context information between multiple entities raises the issue of privacy which has to transparently managed by the user.

### 8.2.2 *Context awareness*

The context awareness principe enables taking automatic decisions either directly with the vehicle embedded system or as remote solution. However the integration of sensoric in the automatisation of tasks, like the selection of music tracks, which have been for long a driver privilege raises new functional and technical issues.

#### 8.2.2.1 *User control*

The full automatisation of user tasks should not completely withdraw the freedom of the user to control the media in the vehicle. Reducing the complexity of infotainment ergonomics will have to avoid reducing the transparency to the user.

#### 8.2.2.2 *Semantic sensoric*

The sensor values that we use in our reasoning framework are not standardised yet. This would make the collaboration with external 'context reasoning' services difficult. As Kaltz [2006] noticed: "Supporting arbitrary sensing in a general-purpose, unifying approach is problematic. Particularly, there are no existing Web standards specifying how environment information is defined and how a Web client would communicate this information to the Web server.". There are actually already standardisation efforts around the topic of 'semantic sensoric'. Ni *et al.* [2009] have identified key subjects: semantic sensory data modeling, semantic-based data dissemination, semantic query processing. An architecture and reasoning methods for SemSOS have been proposed by Henson *et al.* [2009] and could be a basis for further developments.

## 8.3 OUTLOOK

The use of infotainment to stave off user sleepiness or any sort of driving miss-behaviour is an appealing functionality. However one should not forget that vehicle infotainment is often take for what it is : A nice way to entertain and take pleasure to drive and therefore car manufacturers will continue integrating online content. More generally, the tendency to reduce the cognitive load of growingly complex infotainment systems using automatization through context-sensitivity is a very promising innovation path. However, what makes on-demand infotainment so interesting and fascinating to the audience is that people can interact with it. Indeed, the user control over such systems needs to remain transparent and fair otherwise people may prefer to keep their old CD player.

# PUBLICATIONS

We have present several publications during this thesis.

## INTERNATIONAL CONFERENCES AND WORKSHOPS

- Hahn, Clemens; Turlier, Stéphane; Liebig, Thorsten ; Gebhardt, Sascha ; Rölle, Christopher. *Metadata aggregation for personalized music playlists.* USAB'10, HCI in Work & Learning, Life & Leisure, 6th Symposium of the WG HCI&UE of the Austrian Computer Society November 4-5, 2010, Carinthia, Austria - Also published in LNCS, 2010, Volume 6389/2010 , pp 427-442.
- Turlier, Stéphane; Hahn, Clemens; Gebhardt, Sascha. *Browsing online music catalogs in a vehicle: connecting automotive user interfaces with the world wide web.* MCMC'10, ACM International Workshop on Mobile Cloud Media Computing, October 29, 2010, Firenze, Italy , pp 53-58.
- Turlier, Stéphane;Huet, Benoit;Helbig, Thomas; Vögel, Hans-Jörg. *Aggregation and personalization of infotainment, an architecture illustrated with a collaborative scenario.* 8th International Conference on Knowledge Management and Knowledge Technologies, September 4th, 2008, Graz, Austria.

## INTERNATIONAL PATENT

PA 2010029929 DE. Hofer Marc (40%), Turlier Stéphane (40%), Helbig Thomas (20%). *Interoperable Multimedia-Mobilitätslösung.*

# BIBLIOGRAPHY

01net [2006]. *01net. - Les ratés de l'électronique embarquée.*
**URL:** *http://www.01net.com/article/326859.html*

E. Allamanche, *Content-based Identification of Audio Material Using MPEG-7 Low Level Description*. In *ISMIR* (2001).
**URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2001.html#Allamanche01*

C. Anderson, *The long tail: Why the future of business is selling less of more* (Hyperion Books, 2008). ISBN 1401309666.

Apple [2011]. *iTunes - Podcasts - Making a Podcast*.
**URL:** *http://www.apple.com/itunes/podcasts/specs.html#categories*

J. Aucouturier and F. Pachet [2003]. *Representing Musical Genre: A State of the Art*. *Journal of New Music Research*, vol. 32(1):pp. 83–93.

J. Ausubel, C. Marchetti and P. Meyer [2005]. *Toward Green Mobility: the evolution of transport*.
**URL:** *http://phe.rockefeller.edu/green_mobility/*

C. Baccigalupo and E. Plaza, *A Case-Based Song Scheduler for Group Customised Radio*. In S. B. . Heidelberg, ed., *Case-Based Reasoning Research and Development*, vol. Volume 4626/2007 (2007).
**URL:** *http://www.springerlink.com/content/a8w5glg644418045/*

B. Bederson [2000]. *Fisheye Menus*.
**URL:** *http://www.cs.umd.edu/hcil/fisheyemenu/*

A. Berenzweig, B. Logan, D. P. W. Ellis and B. P. W. Whitman [2004]. *A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures*. *Computer Music Journal*, vol. 28(2):pp. 63–76. ISSN 0148-9267.

A. Biffen [2008]. *My Brief History of Mobile Internet*. Website [Last accessed: 21.10.2010] http://connectwirelessweb.com/A-Brief-History-of-Mobile-Internet.htm.

Y. Blanco-Fernández, J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, J. García-Duque, A. Fernández-Vilas, R. Díaz-Redondo and J. Bermejo-Muñoz [2008]. *A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems*. *Knowledge-Based Systems*, vol. 21(4):pp. 305–320.

F. Bobillo and U. Straccia, *fuzzyDL: An expressive fuzzy description logic reasoner*. In *IEEE International Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence)*, pp. 923–930 (2008).

C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber and L. Tanca, *A Data-oriented Survey of Context Models*. vol. 36 (2007).

C. Borgelt, F. Klawonn, R. Kruse and D. Nauck, *Neuro-Fuzzy-Systemen* (2003).

B. Bouchon-Meunier [2003]. *La logique floue*.

M. Broy, M. Gleirscher, P. Kluge, W. Krenzer, S. Merenda and D. Wild [2009]. *Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description*. Tech. Rep..

J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang and X. He, *Music recommendation by unified hypergraph: combining social media information and music content*. In *Proceedings of the international conference on Multimedia*, pp. 391–400 (ACM, 2010).

M. Burmester, R. Graf, J. Hellbrück and A. Meroth, *Usability – Der Mensch im Fahrzeug*, vol. Infotainmentsysteme im Kraftfahrzeug, chap. 8, pp. 321–351 (Friedrich Vieweg & Sohn Verlag, 2008), vieweg praxiswissen edn..

S. Card, T. Moran and A. Newell, *The psychology of human-computer interaction* (CRC, 1983). ISBN 0898598591.

O. Celma [2008]. *Music Recommendation and Discovery in the Long Tail*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain.
**URL:** *http://mtg.upf.edu/ ocelma/PhD/doc/ocelma-thesis.pdf*

D. L. Chao, J. Balthrop and S. Forrest, *Adaptive radio: achieving consensus using negative preferences*. ACM SIGGROUP conference on Supporting group work (ACM, 2005).
**URL:** *http://doi.acm.org/10.1145/1099203.1099224*

S. Chekir [2007]. *Design and Implementation of a Service Oriented Architecture for GUI-dependent Vehicle Services*. Master's thesis, Technische Universität München Lehrstuhl für Kommunikationsnetze.

G. Chen and D. Kotz [2000]. *A Survey of Context-Aware Mobile Computing Research*. Tech. Rep. TR2000-381, Dept. of Computer Science, Dartmouth College.
**URL:** *http://citeseer.ist.psu.edu/chen00survey.html*

Y.-X. Chen and A. Butz, *Musicsim: integrating audio analysis and user feedback in an interactive music browsing ui*. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 429–434 (ACM, New York, NY, USA, 2009). ISBN 978-1-60558-168-2.

S. Cunningham, M. Jones and S. Jones, *Organizing digital music for use: an examination of personal music collections*. In *Proceedings of the 5th International Symposium on Music Information Retrieval* (Citeseer, 2004).

A. K. Dey and G. D. Abowd, *Towards a better understanding of context and context-awareness*. In *Computer Human Intraction 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness* (2000).
**URL:** *ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf*

N. Dibben and V. Williamson [2007]. *An exploratory survey of in-vehicle music listening*. *Psychology of Music*, vol. 35(4):p. 571. ISSN 0305-7356.

T. A. Dingus, M. C. Hulse, J. F. Antin and W. W. Wierwille [1989]. *Attentional demand requirements of an automobile moving-map navigation system*. *Transportation Research Part A: General*, vol. 23(4):pp. 301 – 315. ISSN 0191-2607.
**URL:** *http://www.sciencedirect.com/science/article/B6X3B-4697TJK-3J/2/2bc782e3851af8e7e5933f843dc1882b*

S. Dornbush, A. Joshi, Z. Segall and T. Oates, *A Human Activity Aware Learning Mobile Music Player*. In *Proceeding of the 2007 conference on Advances in Ambient Intelligence*, pp. 107–122 (IOS Press, Amsterdam, The Netherlands, The Netherlands, 2007). ISBN 978-1-58603-800-7.

A. Ehn, M. Hult, F. Niemela, L. Strigeus and G. Kreitz [2007]. *Peer to Peer Streaming of Media Content*.

I. Ehreke [2008]. *Analyse der zielwahl im motorisierten individualverkehr - eine Betrachtung deer Auftretungshäufigkeiten routinisierte Fahrten*. Master's thesis.

T. Exchange [2005]. *Federal Telework: No Free RideProgress Requires Investment and Awareness, but Telework Delivers Significant Opportunity for Federal Government*.

A. Ferrara, L. A. Ludovico, S. Montanelli, S. Castano and G. Haus [2006]. *A semantic web ontology for context-based classification and retrieval of music resources*. *ACM Trans. Multimedia Comput. Commun. Appl*, vol. 2.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee [1999]. *Hypertext Transfer Protocol – HTTP/1.1*.
**URL:** *http://www.ietf.org/rfc/rfc2616.txt*

R. T. Fielding [2000]. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. thesis, University of California, Irvine.
**URL:** *http://www.ics.uci.edu/ fielding/pubs/dissertation/top.htm*

N. Freed and N. Borenstein [1996]. *RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*.
**URL:** *http://tools.ietf.org/html/rfc2046*

G. Furnas [1999]. *The FISHEYE view: A new look at structured files*. *Readings in information visualization: using vision to think*, pp. 312–330.

D. Garlan and D. Perry [1995]. *Introduction to the special issue on software architecture*. *IEEE Transactions on Software Engineering*, vol. 21(4):pp. 269–274. ISSN 0098-5589.

S. Gebhardt [2010]. *Entwicklung eines Anzeige- und Bedienkonzepts für die Verwendung mehrerer großvolumiger Musikdatenquellen im Fahrzeug*. Master's thesis, Ludwig Maximilian Universität, Institut für Informatik.

M. Geiger [2003]. *Berührungslose Bedienung von Infotainment-Systemen im Fahrzeug*. Ph.D. thesis, Technische Universität München.

G. Geiser [1985]. *Man Machine Interaction in Vehicles*. *ATZ*, vol. 87:pp. 74–77.

GFU [2010]. *CE im Auto*.
**URL:** *http://www.gfu.de/home/historie/autoradio.xhtml*

W. T. Glaser, T. B. Westergren, J. P. Stearns and J. M. Kraft [2006]. *Consumer item matching method and system*.

S. M. Global [2009]. *Network Status 2009: The European CDN Market*.
**URL:** *http://www.StreamingMediaGlobal.com/Articles/ReadArticle.aspx?ArticleID=65604&PageNum=2*

O. Goussevskaia, M. Kuhn and R. Wattenhofer, *Exploring music collections on mobile devices.* In G. H. ter Hofte, I. Mulder and B. E. R. de Ruyter, eds., *Mobile HCI*, ACM International Conference Proceeding Series, pp. 359–362 (ACM, 2008). ISBN 978-1-59593-952-4. **URL:** *http://dblp.uni-trier.de/db/conf/mhci/mhci2008.html#GoussevskaiaKW08*

S. Graf, W. Spiessl, A. Schmidt, A. Winter and G. Rigoll, *In-car interaction using search-based user interfaces*. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 1685–1688 (ACM, 2008).

P. Green, *Estimating compliance with the 15-second rule for driver-interface usability and safety*. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 43, pp. 987–991 (Human Factors and Ergonomics Society, 1999). ISSN 1071-1813.

E. Guaus [2009]. *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. Ph.D. thesis, Universitat Pompeu Fabra.

C. Hahn [2010]. *Anbieterübergreifende Metamodellierung von Multimediainhalten zur Generierung und Präsentation von Playlists im Fahrzeug*. Master's thesis, Universität Ulm.

C. Hahn, S. Turlier, T. Liebig, S. Gebhardt and C. Roelle [2010]. *Metadata Aggregation for Personalized Music Playlists. HCI in Work and Learning, Life and Leisure*, pp. 427–442.

J. Haitsma and T. Kalker, *A Highly Robust Audio Fingerprinting System.* In *ISMIR* (2002). **URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2002.html#HaitsmaK02*

M. Hassenzahl [2004]. *The interplay of beauty, goodness, and usability in interactive products. Human-Computer Interaction*, vol. 19(4):pp. 319–349. ISSN 0737-0024.

C. Henson, J. Pschorr, A. Sheth and K. Thirunarayan [2009]. *SemSOS: Semantic sensor observation service.*

S. Hoch, F. Althoff and G. Rigoll [2007]. *The Connected Drive Context Server-flexible Software Architecture for a Context Aware Vehicle. Advanced Microsystems for Automotive Applications*.

F. Hoffmann [2006]. *Metadaten für Multimedia in Überblick*. Tech. Rep..

T. Hofmann [2004]. *Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst.*, vol. 22(1):pp. 89–115. ISSN 1046-8188.

F. Huber [2008]. *Entwicklung eines Frameworks zur transparenten Bereitstellung von Kontextinformationen für Fahrerinformationen und Fahrerassistenzsysteme*. Master's thesis.

S. Hummel [2008]. *Akzeptanzentwicklung bei multimedialen Bedienkonzepten*. Ph.D. thesis, Lehrstuhl für Ergonomie der Technischen Universität München.

T. C. N. I. E. C. (IEC) [1997]. *IEC 1131 - Programmable Controllers*. Tech. Rep..

IFPI [2008]. *Digital Music Report 2008 – Revolution Innovation Responsibility*.

— [2009]. *Digital Music Report 2009 – New Business Models for a Changing environment.*

ISO [1992]. *Passenger car radio connections.*

— [2006]. *Information technology – MPEG systems technologies – Part 1: Binary MPEG format for XML.*

I. Iso [1986]. *Information processing-Text and office systems – Standard Generalized Markup Language (SGML).*

M. Jeon, B. K. Davison, M. A. Nees, J. Wilson and B. N. Walker, *Enhanced auditory menu cues improve dual task performance and are preferred with in-vehicle technologies.* In *AutomotiveUI '09: Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 91–98 (ACM, New York, NY, USA, 2009). ISBN 978-1-60558-571-0.

J. W. Kaltz [2006]. *An Engineering Method for Adaptive, Context-aware Web Applications.* Ph.D. thesis, Universität Duisburg-Essen.

T. Kastner, E. Allamanche, J. Herre, O. Hellmuth, M. Cremer and H. Grossmann [2002]. *MPEG-7 Scalable Robust Audio Fingerprinting.*

C. Katsma and T. Spil [2010]. *A taxonomy of digital music services. AMCIS 2010 Proceedings*, p. 559.

F. G. Kazasis, N. Moumoutzis, N. Pappas, A. Karanastasi and S. Christodoulakis, *Designing Ubiquitous Personalized TV-Anytime Services.* In *University of Maribor*, pp. 136–149 (Press, 2003).

Z. I. Kazi-Aoul [2008]. *Une architecture orientée services pour la fourniture de documents multimédia composés adaptables.* Ph.D. thesis, Telecom ParisTech.

M. Keidl [2004]. *Metadata Management and Context-based Personalization in Distributed Information Systems.* Ph.D. thesis, Technische Universität München.
**URL:** *http://tumb1.biblio.tu-muenchen.de/publ/diss/in/2004/keidl.pdf*

H.-G. Kim, N. Moreau and T. Sikora, *Low-Level Descriptors*, chap. 2 (2005).

J. Kim and N. Belkin, *Categories of music description and search terms and phrases used by non-music experts.* In *Proceedings of the Third International Conference on Music Information Retrieval: ISMIR*, pp. 209–214 (Citeseer, 2002).

G. Kreitz and F. Niemela, *Spotify–Large Scale, Low Latency, P2P Music-on-Demand Streaming.* In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, pp. 1–10 (IEEE, 2010).

G. Krump, *Sound Systems in Cars* (2008).

P. Lamere and D. Eck, *Using 3D Visualization to explore and discover Music* (2007).

P. C. B. T. C. Lansdown [2000]. *E-Distraction: The Challenges for Safe and Usable Internet Services in Vehicles.*

J. Law-To, G. Grefenstette and J. Gauvain, *VoxaleadNews: robust automatic segmentation of video into browsable content*. In *Proceedings of the seventeen ACM international conference on Multimedia*, pp. 1119–1120 (ACM, 2009).

H. Lee, *Issues in Designing Novel Applications for Emerging Multimedia Technologies*. In G. Leitner, M. Hitz and A. Holzinger, eds., *HCI in Work and Learning, Life and Leisure*, vol. 6389 of *Lecture Notes in Computer Science*, pp. 411–426 (Springer, 2010). ISBN 978-3-642-16606-8.

J. Lee, B. Caven, S. Haake and T. Brown [2001]. *Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers' attention to the roadway*. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 43(4):p. 631. ISSN 0018-7208.

S. Lee and K. Chung [2008]. *Buffer-driven adaptive video streaming with TCP-friendliness*. *Comput. Commun.*, vol. 31(10):pp. 2621–2630. ISSN 0140-3664.

T. B. Lee, *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor* (Britain: Orion Business, 1999).

M. Lesaffre [2006]. *Music Information Retrieval - Conceptual Framework, Annotation and User Behaviour*. Ph.D. thesis, Faculty of Arts and Philosophy, Department of Art, Music and Theatre Sciences Ghent University.

M. Lesaffre, M. Leman, K. Tanghe, B. D. Baet, H. D. Meyer and J.-P. Martens, *User-Dependent Taxonomy of Musical Features As a Conceptual Framework for Musical Audio-Mining Technology* (Stockholm Music Acoustics Conference, Stockholm, Sweden, 2003).

V. Levenshtein, *Binary Codes Capable of Correcting Deletions, Insertions and Reversals*. In *Soviet Physics Doklady*, vol. 10 (1966).

H. Lieberman [2009]. *User interface goals, AI opportunities*. *AI Magazine*, vol. 30(4):p. 16. ISSN 0738-4602.

A. S. Lillie [2008]. *MusicBox: Navigating the space of your music*. Master's thesis, Massachussets Institute of Technology.

S. Loeb [1992]. *Architecting personalized delivery of multimedia information*. *Commun. ACM*, vol. 35(12):pp. 39–47. ISSN 0001-0782.

S. MacKenzie, *Mobile text entry using three keys*. In *Proceedings of the second Nordic conference on Human-computer interaction*, NordiCHI '02, pp. 27–34 (ACM, New York, NY, USA, 2002). ISBN 1-58113-616-1. **URL:** *http://doi.acm.org/10.1145/572020.572025*

B. Manjunath, P. Salembier and T. Sikora, eds., *Introduction to MPEG-7: multimedia content description interface* (John Wiley & Sons Inc, 2002). ISBN 0471486787.

G. Matthews, T. J. Sparkes and H. M. Bygrave [1996]. *Attentional Overload, Stress, and Simulated Driving Performance*. *Human Performance*, vol. 9.

D. McQuail, *Audience analysis* (Sage Publications, Inc, 1997). ISBN 0761910026.

P. Melville and V. Sindhwani [2010]. *Recommender Systems*.

A. Meroth and B. Tolg, *Infotainmentsysteme im Kraftfahrzeug* (2008).

MPEG [1993]. *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*.

— [1995]. *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*.

L. Ni, Y. Zhu, J. Ma, Q. Luo, Y. Liu, S. Cheung, Q. Yang, M. Li and M. Wu [2009]. *Semantic Sensor Net: an extensible framework*. *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 4(3):pp. 157–167.

NISO [2004]. *Understanding metadata*. National Information Standards Organization. ISBN 1-880124-62-9.
**URL:** *http://www.niso.org/standards/resources/UnderstandingMetadata.pdf*

O. Noppens, T. Liebig, P. Schmidt, M. Luther and M. Wagner [2007a]. *MobiXpl a svg-based mobile user interface for semantic service discovery*.
**URL:** *http://www.svgopen.org/2007/papers/MobiXpl/index.html*

O. Noppens, M. Luther, T. Liebig, M. Wagner and M. Paolucci, *Ontology-supported Preference Handling for Mobile Music Selection*. In *Conference Paper, ECAI*, vol. 6 (Citeseer, 2007b).

M. O'Connor, D. Cosley, J. A. Konstan and J. Riedl, *PolyLens: A Recommender System for Groups of Users*. In *ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pp. 199–218 (Kluwer Academic Publishers, Norwell, MA, USA, 2001). ISBN 0-7923-7162-3.

O. Omojokun, M. Genovese and C. Isbell, *Impact of user context on song selection*. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pp. 897–900 (ACM, New York, NY, USA, 2008). ISBN 978-1-60558-303-7.

A. Oniszczak and I. S. MacKenzie, *A comparison of two input methods for keypads on mobile devices*. In *Proceedings of the third Nordic conference on Human-computer interaction*, NordiCHI '04, pp. 101–104 (ACM, New York, NY, USA, 2004). ISBN 1-58113-857-1.
**URL:** *http://doi.acm.org/10.1145/1028014.1028030*

C. Osgood, *The measurement of meaning* (Univ of Illinois Pr, 1975). ISBN 0252745396.

F. Pachet and D. Cazaly, *A Taxonomy of Musical Genres*. In *Content-Based Multimedia Information Access Conference (RIAO) proceedings* (2000).

E. Pampalk [2001]. *Island of Music - Analysis, Organisation and Visualisation of Music Archives*. Master's thesis, Institut für Softwaretechnik und Interaktive Systeme, Technische Universität Wien.

— [2006]. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. Ph.D. thesis, Vienna University of Technology, Vienna, Austria.
**URL:** *http://www.ofai.at/ elias.pampalk/publications/pampalk06thesis.pdf*

E. Pampalk and M. Gasser, *An Implementation of a Simple Playlist Generator Based on Audio Similarity Measures and User Feedback.* In *Proceedings of 7th International Conference on Music Information Retrieval*, pp. 389–390 (Victoria, Canada, 2006).
**URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2006.html#PampalkG06a*

E. Pampalk and M. Goto, *MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling.* In *ISMIR*, pp. 367–370 (2006).
**URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2006.html#PampalkG06*

M.-H. Park, J.-H. Hong and S.-B. Cho, *Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices.* In *Ubiquitous Intelligence and Computing*, vol. 4611 of *Lecture Notes in Computer Science*, pp. 1130–1139 (Springer Berlin / Heidelberg, 2007). ISBN 978-3-540-73548-9. ISSN 0302-9743 (Print) 1611-3349 (Online).
**URL:** *http://www.springerlink.com/content/j10232118932324j/*

A. Passant [2010]. *dbrec–Music Recommendations Using DBpedia*. *The Semantic Web–ISWC 2010*, pp. 209–224.

A. Passant and S. Decker [2010]. *Hey! Ho! Let's Go! Explanatory Music Recommendations with dbrec*. *The Semantic Web: Research and Applications*, pp. 411–415.

D. Patrascu [2009a]. *BMW to Help Drivers with AI-Like ILENA System*.
**URL:** *http://www.autoevolution.com/news/bmw-to-help-drivers-with-ai-like-ilena-system-4430.html*

— [2009b]. *ILENA Confirmed by BMW*.
**URL:** *http://www.autoevolution.com/news/ilena-confirmed-by-bmw-4615.html*

C. Pautasso, *REST vs. SOAP: Making the Right Architectural Decision*. In *SOA Symposium*, pp. 2009–01 (Citeseer, 2008a).

— [2008b]. *RESTful Web Services*. *Online at" http://www. iks. inf. ethz. ch/education/ss08/ws_soa/RESTfulWebServices. pdf*.

S. Pauws and B. Eggen, *PATS: Realization and user evaluation of an automatic playlist generator*. In *In ISMIR*, pp. 222–230 (2002).

S. Pauws and S. van de Wijdeven, *User Evaluation of a New Interactive Playlist Generation Concept*. In *Proceedings of 6th International Conference on Music Information Retrieval*, pp. 638–643 (London, UK, 2005).
**URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2005.html#PauwsW05*

I. Peters and K. Weller [2008]. *Tag gardening for folksonomy enrichment and maintenance*. *Webology*, vol. 5(3).

M. A. Pettitt [2008]. *Visual demand evaluation methods for in-vehicle interfaces*. Ph.D. thesis, The University of Nottinghan.

F. P. D. Piotet, *Comment le web change le monde* (Pearson Education France, 2008).

H.-C. Quelle and T. Kusche [2006]. *RadioText Plus – a new enhancement to the RDS RadioText service*. *EBU Technical Review*.

Y. Raimond [2008]. *A Distributed Music Information System*. Ph.D. thesis, Queen Mary, University of London.

J. Rayport and J. Sviokla [1996]. *Exploiting the Virtual Value Chain. The McKinsey Quarterly*, (1):pp. 21–22.

S. Reddy and J. Masciaa [2006]. *Lifetrak: music in tune with your life*. p. 34.

C. Rölle [2007]. *Entwicklung einer kartographischen Interaktionsmetapher für große Datenmengen im Automobil*. Master's thesis, Technische Universität München.

T. A. Runkler, *Data Mining* (Vieweg + Teubner, Wiesbaden, 2010). ISBN 978-3-8348-0858-5.

J. Russell [1980]. *A circumplex model of affect*. *Journal of personality and social psychology*, vol. 39(6):pp. 1161–1178. ISSN 0022-3514.

J. Russell, M. Lewicka and T. Niit [1989]. *A cross-cultural study of a circumplex model of affect*. *Journal of personality and social psychology*, vol. 57(5):pp. 848–856.

C. Salzmann [2009]. *Automotive Software – Methoden und Technologien*.

N. Sarter [2006]. *Multimodal information presentation: Design guidance and research challenges*. *International journal of industrial ergonomics*, vol. 36(5):pp. 439–445. ISSN 0169-8141.

B. Sarwar, G. Karypis, J. Konstan and J. Reidl, *Item-based collaborative filtering recommendation algorithms*. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295 (ACM, 2001). ISBN 1581133480.

W. N. Schilit [1995]. *A system architecture for context-aware mobile computing*. Ph.D. thesis, New York, NY, USA.

B. Schuller, J. Dorfner and G. Rigoll [2010]. *Determination of nonprototypical valence and arousal in popular music: features and performances*. *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010:p. 5. ISSN 1687-4714.

B. Schwartz, *The paradox of choice: Why more is less* (Harper Perennial, 2005). ISBN 0060005696.

W. Seufert and M. Ehrenberg [2007]. *Microeconomic Consumption Theory and Individual Media Use: Empirical Evidence from Germany*. *Journal of Media Business Studies*, vol. 4(3):pp. 21–39.

J. Sloboda, S. O'Neill and A. Ivaldi [2001]. *Functions of music in everyday life: An exploratory study using the Experience Sampling Method. Musicae Scientiae*. ISSN 1029-8649.

A. Stahl [2003]. *Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning*. Ph.D. thesis, University of Kaiserslautern.

D. Strayer, F. Drews, W. Johnston *et al.* [2003]. *Cell phone-induced failures of visual attention during simulated driving*. *Journal of Experimental Psychology Applied*, vol. 9(1):pp. 23–32. ISSN 1076-898X.

K. Swearing and R. Sinha, *Interaction Design for Recommender Systems*. In *Designing Interactive Systems* (2002).

Technofriends [2008]. *Understanding Content Delivery Networks*. **URL:** *http://technofriends.in/2008/12/14/understanding-content-delivery-networks/*

R. Thayer, *The biopsychology of mood and arousal* (Oxford University Press, USA, 1989). ISBN 0195068270.

The National Archives [2007]. *The Soundex Indexing System*. **URL:** *http://www.archives.gov/research/census/soundex.html*

M. Tonnis, V. Broy and G. Klinker [2006]. *A Survey of Challenges Related to the Design of 3D User Interfaces for Car Drivers*.

R. Troncy [2010]. *Bringing the iptc news architecture into the semantic web*. *The Semantic Web-ISWC 2008*, pp. 483–498.

S. Turlier, C. Hahn and S. Gebhardt, *Browsing online music catalogs in a vehicle: connecting automotive user interfaces with the world wide web*. In *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing*, pp. 53–58 (ACM, 2010).

S. Turlier, B. Huet, T. Helbig and H.-J. Vögel, *Aggregation and Personalization of Infotainment, An Architecture Illustrated with a Collaborative Scenario* (2008). Graz, Austria. **URL:** *http://www.eurecom.fr/util/publidownload.fr.htm?id=2677*

G. Urdaneta, G. Pierre and M. V. Steen [2009]. *A survey of DHT security techniques*. *ACM Computing Surveys*.

D. Vallet, P. Castells, M. Fernandez, P. Mylonas and Y. Avrithis [2007]. *Personalized Content Retrieval in Context Using Ontological Knowledge*. *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17(3):pp. 336–346. ISSN 1051-8215.

F. Vignoli, *Digital Music Interaction Concepts: A User Study*. In *ISMIR* (2004). **URL:** *http://dblp.uni-trier.de/db/conf/ismir/ismir2004.html#Vignoli04*

D. Vodopivec [2010]. *On-demand Television combined with non-real-time Peer-to-Peer Content Delivery for Television Content Providers*. Master's thesis.

A. von Hessling [2004]. *Ontology based profile matching on mobile service*. Tech. Rep..

W3C [2008]. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. **URL:** *http://www.w3.org/TR/REC-xml/*

Q. Wang, W. Balke, W. Kießling and A. Huhn [2004]. *P-news: Deeply personalized news dissemination for mpeg-7 based digital libraries*. *Research and Advanced Technology for Digital Libraries*, pp. 256–268.

D. Weiß [2009]. *Kontext-abhängige Personalisierung multimedialer Inhalte auf mobilen Endgeräten*. Ph.D. thesis, Ludwig Maximilian Universität Institut für Informatik.

N. Weißenberg, R. Gartmann and A. Voisard [2006]. *An ontology-based approach to personalized situation-aware mobile service supply*. *Geoinformatica*, vol. 10(1):pp. 55–90.

K. West, S. Cox and P. Lamere, *Incorporating machine-learning into music similarity estimation*. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pp. 89–96 (ACM, New York, NY, USA, 2006). ISBN 1-59593-501-0.

Wikipedia [2010a]. *Peer-to-peer*.
**URL:** *http://en.wikipedia.org/wiki/Peer_to_peer*

— [2010b]. *Semantic differential*.
**URL:** *http://en.wikipedia.org/wiki/Semantic_differential*

K. Wilkinson, C. Sayers, H. Kuno and D. Reynolds, *Efficient RDF Storage and Retrieval in Jena2*. In *Proc. First International Workshop on Semantic Web and Databases* (2003).
**URL:** *http://www.cs.uic.edu/ ifc/SWDB/papers/Wilkinson_etal.pdf*

K. Young and M. Regan [2007]. *Driver distraction: A review of the literature. Distracted driving. Sydney, NSW: Australasian College of Road Safety*, pp. 379–405.

S.-T. Yuan and Y. W. Tsao [2003]. *A recommendation mechanism for contextualized mobile advertising*. *Expert Systems with Applications*, vol. 24(4):pp. 399 – 414. ISSN 0957-4174.
**URL:** *http://www.sciencedirect.com/science/article/B6V03-47RBCJT-1/2/6f5aa281bc7764207a34abd32dbb1782*

J. Zaletelj, T. Pozrl, M. Kunaver, T. Mlakar, M. Meza, M. Pogacnik, A. Pütz and M. Güde [2007]. *Report on user models and content selection methods*. Tech. Rep..

A. Zimmermann, M. Specht and A. Lorenz [2005]. *Personalization and Context Management*. *User Modeling and User-Adapted Interaction*, vol. 15(3-4):pp. 275–302. ISSN 0924-1868.
**URL:** *http://dx.doi.org/10.1007/s11257-005-1092-2*

H. Zwahlen, C. Adams and D. D. Bald, *Safety aspects of CRT touch panel controls in automobiles*. In *Vision in vehicles–II: proceedings of the Second International Conference on Vision in Vehicles, Nottingham, UK, 14-17 September 1987*, p. 335 (North Holland, 1988). ISBN 044470423X.

H. Zwahlen and D. DeBald, *Safety aspects of sophisticated in-vehicle information displays and controls*. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 30, pp. 256–260 (Human Factors and Ergonomics Society, 1986). ISSN 1071-1813.

# Part III

# APPENDIX

# APPENDIX

---

## FORMAL DEFINITION OF DISTANCE FUNCTIONS

Distances functions[1] can measure how far apart objects represented by mathematical values are one from another. An application $d$ computes the distance for every $x$ and $y$ vectors of size $n$.

$$(x,y) \in \mathbb{R}^n \times \mathbb{R}^n \longmapsto d(x,y) \in \mathbb{R}$$

Similarity functions verify the following conditions:

$$d(x,y) = d(y,x) \quad d(x,y) = 0 \Leftrightarrow x = y \quad d(x,y) + d(y,z) \geqslant d(x,z)$$

*Some examples of distance measures*

- Manhattan

$$d(x,y) = \sum_{i=1}^{n} |x^{(i)} - y^{(i)}|$$

- Euclidian

$$d(x,y) = \sqrt{\sum_{i=1}^{n} |x^{(i)} - y^{(i)}|^2}$$

- Minkowski

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x^{(i)} - y^{(i)}|^p}$$

- Tchebychev

$$d(x,y) = \lim_{p \to \infty} \sqrt[p]{\sum_{i=1}^{n} |x^{(i)} - y^{(i)}|^p}$$

## FORMAL DEFINITION OF SIMILARITY FUNCTIONS[1]

Similarity measures quantify the similarity of pairs of objects, using the description vectors of those objects. An application $s$ computes the similarity for every $x$ and $y$ vectors of size $n$.

$$(x,y) \in \mathbb{R}^n \times \mathbb{R}^n \longmapsto s(x,y) \in \mathbb{R}$$

Similarity functions verify the following conditions:

$$s(x,y) = s(y,x)$$
$$s(x,y) \geqslant 0$$
$$s(x,x) \geqslant s(x,y)$$

---

[1] most of the examples here are taken from Runkler [2010, p.11-13]

*Some examples of similarity measures*

- Cosinus

$$s(x,y) = \frac{\sum_{i=1}^{n} x^{(i)} \cdot y^{(i)}}{\sqrt{\sum_{i=1}^{n} \left(x^{(i)}\right)^2 \cdot \sum_{i=1}^{n} \left(y^{(i)}\right)^2}}$$

- Pearson

$$s(x,y) = \frac{\sum_{i=1}^{n} (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)^2 \cdot \sum_{i=1}^{n} \left(y^{(i)} - \bar{y}\right)^2}}$$

  where $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$

- Overlap

$$s(x,y) = \frac{\sum_{i=1}^{n} x^{(i)} y^{(i)}}{\min\left(\sum_{i=1}^{n} \left(x^{(i)}\right)^2, \sum_{i=1}^{n} \left(y^{(i)}\right)^2\right)}$$

- Dice

$$s(x,y) = \frac{2 \sum_{i=1}^{n} x^{(i)} \cdot y^{(i)}}{\sum_{i=1}^{n} \left(x^{(i)}\right)^2 + \sum_{i=1}^{n} \left(y^{(i)}\right)^2}$$

- Jaccard

$$s(x,y) = \frac{\sum_{i=1}^{n} x^{(i)} \cdot y^{(i)}}{\sum_{i=1}^{n} \left(x^{(i)}\right)^2 + \sum_{i=1}^{n} \left(y^{(i)}\right)^2 - \sum_{i=1}^{n} x^{(i)} y^{(i)}}$$

## THE ATTRAKDIFF QUESTIONNAIRE FOR THE USER STUDY

This is the questionnaire that was used for the user study according to the AttrakDiff method [Gebhardt, 2010]:

## Angaben zur Person

Alter: ______ Jahre

Geschlecht: ☐ weiblich
☐ männlich

Jährliche Fahrleistung: ☐ < 5.000km
☐ 5.000km-10.000km
☐ > 10.000km

## Im Stand (30 min)

### 00:00 Einleitung

- 7er L ist Prestigefahrzeug: „vorbildlich“ fahren
- Teures Testfahrzeug mit teilweise losen Teilen: ruhig fahren ohne starkes Beschleunigen/Bremsen
- Fahraufgabe hat absolute Priorität
- Für die Aufgaben die gestellt werden besteht keinerlei Zeitdruck
- Bitte „laut denken“; beschreiben, wie das Nutzerinterface erlebt wird
- Semantische Differenziale erklären

### 00:05 Proband Anwendung im Stand ausprobieren lassen (10 min)

### 00:15 Anwendungsstruktur erklären, Feedback geben lassen (10 min)

- Struktur erklären:
  - Zwei Modi: Play, Filter; übereinander angeordnet
  - Filtermodus: Filter können hinzugefügt/entfernt werden, Filterkategorien, Previews
  - Playmodus:
    - Drei Teile: laufender Song, History, Playlisten; nebeneinander angeordnet
    - Bei Playlisten: oberhalb Album, unterhalb ähnliche Musik

## 00:20 Dynamik erklären, Feedback geben lassen (5 min)

- Dynamik erklären:
    - Playlisten und History werden größer, wenn sie in den Fokus rücken
    - Song mit Fokus ist am größten
    - Album/Similar Songs zum gespielten Song erscheinen durch raussliden -> sind im Gegensatz zur Hauptplaylist dynamisch (sowohl optisch als auch konzeptionell)
    - History wird größer und kriegt mehr Platz, wenn sie in den Fokus rückt
    - Bei Moduswechsel wird kurz der Name des Modus in den man wechselt eingeblendet

## 00:25 AttrakDiff (5min)

---

## Ihr Urteil (Bewertung der Anwendung vor der Fahrt)!

Nachfolgend finden Sie Wortpaare, mit deren Hilfe Sie die die Anwendung bewerten können. Sie stellen jeweils extreme Gegensätze dar, zwischen denen eine Abstufung möglich ist.

*Ein Beispiel:*

unsympathisch ☐ ☐ ☐ ☐ ☒ ☐ ☐ sympathisch

*Diese Bewertung bedeutet, dass die Anwendung für Sie eher sympathisch ist.*

Denken Sie nicht lange über die Wortpaare nach, sondern geben Sie bitte die Einschätzung ab, die Ihnen spontan in den Sinn kommt.
Vielleicht passen einige Wortpaare nicht so gut auf die Anwendung, kreuzen Sie aber trotzdem bitte immer eine Antwort an. Denken Sie daran, dass es keine "richtigen" oder "falschen" Antworten gibt - nur Ihre persönliche Meinung zählt!

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| einfach | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | kompliziert |
| hässlich | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | schön |
| praktisch | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | unpraktisch |
| stilvoll | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | stillos |
| voraussagbar | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | unberechenbar |
| minderwertig | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | wertvoll |
| phantasielos | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | kreativ |
| gut | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | schlecht |
| verwirrend | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | übersichtlich |
| lahm | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | fesselnd |

# APPENDIX

## PICTURES OF THE PERSONAL RADIO PROTOTYPE USING THE BMW REMOTE HMI

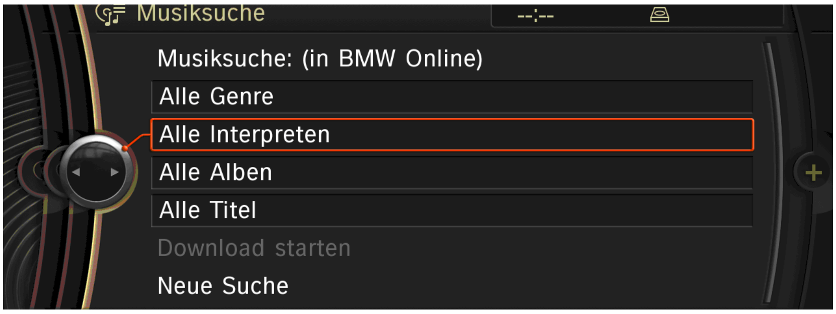


Figure 77: Selection of the search type

Figure 78: Text user input

Figure 79: List result display

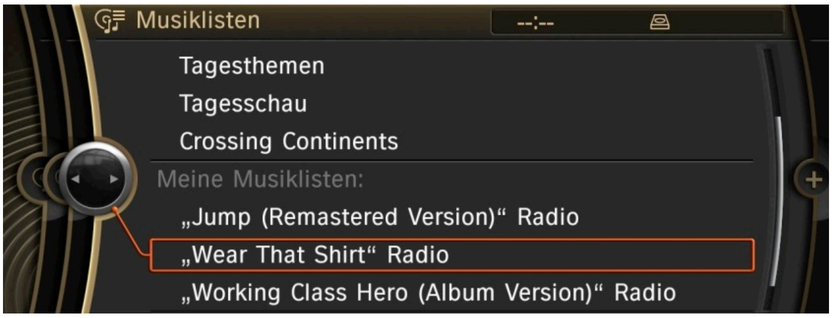


Figure 80: Separation between music and infotainment channels

## SOAP: A LIGHT PROTOCOL FOR WEB SERVICES

### *Exemple of SOAP requests in personal radio*

Listing 4: Renew channel request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<prMsg:personalRadioRequest xmlns:prMsg="http://www.bmw.de/
    PersonalRadio/prMessage">
<prMsg:commonParameters>
<prMsg:requestID>12345678912</prMsg:requestID>
<prMsg:market>DE</prMsg:market>
<prMsg:language>de</prMsg:language>
<prMsg:version>1</prMsg:version>
</prMsg:commonParameters>
<prMsg:operationName>renewChannelURLs</prMsg:operationName>
<prMsg:operationParameters>
<prMsg:operationParameter key="channel">
<channel><channelLite xmlns="">
        <bmwUniqueID>3257</bmwUniqueID>
        <name>dradio-Andruck</name>
 <contentProviderID>4</contentProviderID>
        <type>MANAGED</type>
      </channelLite>
      <channelItemCount xmlns="">30</channelItemCount>
      <description xmlns="">Beitr\"age des Deutschlandradio (
          Deutschlandfunk, Deutschlandradio Kultur)</description>
      <contentItemCollection>
   <coverImageSmallURL>http://localhost:8080/podcast/
       f1d16360d70adebad96d5379c1959f44.55.jpg</coverImageSmallURL>
   <coverImageLargeURL>http://localhost:8080/podcast/
       f1d16360d70adebad96d5379c1959f44.320.jpg</coverImageLargeURL
       >
   <collectionType>MANAGED_PODCAST</collectionType>
   <collectionDescription>Beitr\"age des Deutschlandradio (
       Deutschlandfunk, Deutschlandradio Kultur)</
       collectionDescription>
   <contentObjectList>
     <contentObject>
       <contentObjectLite>
         <providerUniqueID>http://podcast-mp3.dradio.de/podcast
             /2010/05/17/dlf_20100517_1952_06f5dfb0.mp3</
             providerUniqueID>
         <contentProviderID>4</contentProviderID>
         <name1>DLF</name1>
```

```
         <name2>dradio-Andruck</name2>
         <name3>Verband der Heimkehrer - Birgit Schwelling:    "
             Heimkehr — Erinnerung — Integration". ...</name3>
       </contentObjectLite>
       <date1>2010-05-17T19:52:19.000+02:00</date1>
       <genre>Politik</genre>
       <contentObjectType>
         <contentItem>
           <channelID>f1d16360d70adebad96d5379c1959f44</channelID>
           <length>363</length>
           <itemType>PODCAST</itemType>
           <description>Etwa elf Millionen Kriegsheimkehrer mussten
                nach dem Zweiten Weltkrieg in die demokratische
               Gesellschaft Deutschlands integriert werden. Diese
               Personengruppe galt als besonders gef\"ahrlich f\"ur
                die Demokratie, hatte sie doch in der Weimarer
               Republik zu deren erbittersten Gegnern geh\"ort.
               Deutschlandfunk, Andruck</description>
           <downloadFileURL>http://podcast-mp3.dradio.de/podcast
               /2010/05/17/dlf_20100517_1952_06f5dfb0.mp3</
               downloadFileURL>
           <downloadPictureURL>http://localhost:8080/podcast/
               f1d16360d70adebad96d5379c1959f44.320.jpg</
               downloadPictureURL>
           <restrictions>
             <searchable>true</searchable>
             <rented>true</rented>
             <copyProtected>true</copyProtected>
             <leaseEndDate>2100-01-01T00:00:00</leaseEndDate>
             <playCount>-1</playCount>
           </restrictions>
           <downloadSize>2906592</downloadSize>
           <mediaCodec>
             <mimeType>audio/mpeg</mimeType>
           </mediaCodec>
         </contentItem>
       </contentObjectType>
     </contentObject>
        </contentObjectList>
      </contentItemCollection>
      <playerFunctionalDescription xmlns="">
        <viewPlaylist>true</viewPlaylist>
        <viewContentItemFuture>-1</viewContentItemFuture>
        <viewContentItemHistory>-1</viewContentItemHistory>
        <skipContentItem>-1</skipContentItem>
        <displayAlbumArt>true</displayAlbumArt>
        <refreshChannel>true</refreshChannel>
        <timeStampBasedContentPlayback>false</
            timeStampBasedContentPlayback>
        <invisible>false</invisible>
      </playerFunctionalDescription></channel>
</prMsg:operationParameter>
</prMsg:operationParameters>
</prMsg:personalRadioRequest>
```

Listing 5: Synchronize profile request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<prMsg:personalRadioRequest xmlns:prMsg="http://www.bmw.de/
    PersonalRadio/prMessage">
<prMsg:commonParameters>
<prMsg:requestID>12345678912</prMsg:requestID>
<prMsg:market>DE</prMsg:market>
<prMsg:language>de</prMsg:language>
<prMsg:version>1</prMsg:version>
</prMsg:commonParameters>
<prMsg:operationName>synchroniseProfile</prMsg:operationName>
<prMsg:operationParameters>
<prMsg:operationParameter key="profile">
<profile>
<serviceID />
<version />
<listOfQueryChannels />
<favoriteContentItems />
<maxNumberOfFavorites />
<taggedSongList />
<rootLicenceList>
  <contentProviderLicence>
    <contentProviderID />
    <expiryDate />
  </contentProviderLicence>
</rootLicenceList>
<listOfAvailableFunctions>
  <subscribeToManagedChannel>false</subscribeToManagedChannel>
  <unsubscribeFromManagedChannel>true</
      unsubscribeFromManagedChannel>
  <playManagedChannel>true</playManagedChannel>
  <createQueryBasedChannelOnArtist>true</
      createQueryBasedChannelOnArtist>
  <createQueryBasedChannelOnSong>true</
      createQueryBasedChannelOnSong>
  <deleteQueryChannel>true</deleteQueryChannel>
  <playQueryChannel>true</playQueryChannel>
  <browseChannel>true</browseChannel>
  <rentContentItem>true</rentContentItem>
  <rentNumberItemAllowed>-1</rentNumberItemAllowed>
  <rentNumberCollectionsAllowed>-1</rentNumberCollectionsAllowed>
  <displayAlbumList>true</displayAlbumList>
  <accessAlbumSelection>true</accessAlbumSelection>
  <requestAdditionalArtistInfo>true</requestAdditionalArtistInfo>
  <requestAdditionalAlbumInfo>true</requestAdditionalAlbumInfo>
  <requestAdditionalArtistNews>true</requestAdditionalArtistNews>
  <tagContentItem>true</tagContentItem>
  <purchaseContentItem>true</purchaseContentItem>
  <onlineMusicSearch>true</onlineMusicSearch>
  <addContentItemToFavourites>true</addContentItemToFavourites>
  <useAudioNews>true</useAudioNews>
  <audioNewsSearch>true</audioNewsSearch>
  <audioNewsTopNewsList />
  <audioNewsPersonalList />
  <audioNewsPlayerFunctions>
    <playback>true</playback>
    <rateUp>true</rateUp>
    <rateDown>true</rateDown>
    <addFavorites>true</addFavorites>
    <maxNumberOfFavourite>-1</maxNumberOfFavourite>
    <maxNumberOfTimePeriod>-1</maxNumberOfTimePeriod>
```

```
    <skipForward>-1</skipForward>
    <skipBackward>-1</skipBackward>
  </audioNewsPlayerFunctions>
  <contentIdentificationAvailable>true</
      contentIdentificationAvailable>
  <fingerPrintingAvailable>true</fingerPrintingAvailable>
  <supportOtherSources>true</supportOtherSources>
  <timeStampBasedContentSupport>true</timeStampBasedContentSupport>
  <timeStampEntertainmentServer>true</timeStampEntertainmentServer>
</listOfAvailableFunctions>
<consumptionHistory />
<channelBufferLength />
<channelUpdateItemCount />
<lastSuccesfulSynchroniseDate />
<refrenceDateAndTime />
<contentItemUpdateObjectList />
<listOfSubscribedManagedChannels />
<type>
  <userProfile>
    <lisboaID>4712</lisboaID>
  </userProfile>
</type>
</profile>
</prMsg:operationParameter>
<prMsg:operationParameter key="type">SYNCHRONISE</
    prMsg:operationParameter>
</prMsg:operationParameters>
</prMsg:personalRadioRequest>
```

Listing 6: The vehicle environment model ontology (partial version)

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY fzl "http://ontology.bmw.de/fzl#" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<rdf:RDF xmlns="http://ontology.bmw.de/veem#"
     xml:base="http://ontology.bmw.de/veem"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:fzl="http://ontology.bmw.de/fzl#">
    <owl:Ontology rdf:about="http://ontology.bmw.de/veem">
        <dc:language>en</dc:language>
        <rdfs:comment>limited version</rdfs:comment>
        <dc:title>Vehicle Environment Model</dc:title>
        <dc:creator>St&#233;phane Turlier</dc:creator>
```

```
    <owl:imports rdf:resource="http://ontology.bmw.de/fzl"/>
</owl:Ontology>


<owl:AnnotationProperty rdf:about="&dc;creator"/>
<owl:AnnotationProperty rdf:about="&dc;language"/>
<owl:AnnotationProperty rdf:about="&rdfs;comment"/>
<owl:AnnotationProperty rdf:about="&dc;title"/>


<!-- http://ontology.bmw.de/fzl#fmax -->

<owl:DatatypeProperty rdf:about="&fzl;fmax"/>



<!-- http://ontology.bmw.de/fzl#fmidHigh -->

<owl:DatatypeProperty rdf:about="&fzl;fmidHigh"/>



<!-- http://ontology.bmw.de/fzl#fmidLow -->

<owl:DatatypeProperty rdf:about="&fzl;fmidLow"/>



<!-- http://ontology.bmw.de/fzl#fmin -->

<owl:DatatypeProperty rdf:about="&fzl;fmin"/>



<!-- http://www.w3.org/2002/07/owl#topDataProperty -->

<owl:DatatypeProperty rdf:about="&owl;topDataProperty"/>



<!-- http://ontology.bmw.de/fzl#FuzzyVariable -->

<owl:Class rdf:about="&fzl;FuzzyVariable"/>



<!-- http://ontology.bmw.de/veem#Context -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#Context">
    <rdfs:subClassOf rdf:resource="&fzl;FuzzyVariable"/>
</owl:Class>



<!-- http://ontology.bmw.de/veem#DayTime -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#DayTime">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/veem#
        TimeContext"/>
```

```
</owl:Class>



<!-- http://ontology.bmw.de/veem#DestinationDistance -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#
     DestinationDistance">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/veem#
         MotionContext"/>
</owl:Class>



<!-- http://ontology.bmw.de/veem#MotionContext -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#MotionContext
     ">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/veem#
         Context"/>
</owl:Class>



<!-- http://ontology.bmw.de/veem#TimeContext -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#TimeContext">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/veem#
         Context"/>
</owl:Class>



<!-- http://ontology.bmw.de/veem#Traffic -->

<owl:Class rdf:about="http://ontology.bmw.de/veem#Traffic">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/veem#
         MotionContext"/>
</owl:Class>


<!-- http://ontology.bmw.de/veem#AFTERNOON -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    AFTERNOON">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
         "/>
    <fzl:fmin rdf:datatype="&xsd;double">12</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">14</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">17</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">19</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#CONGESTIONNED -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    CONGESTIONNED">
```

```
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#Traffic
        "/>
    <fzl:fmin rdf:datatype="&xsd;double">31</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">33</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">65</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">67</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#EVENING -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    EVENING">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
        "/>
    <fzl:fmin rdf:datatype="&xsd;double">18</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">20</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">22</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">24</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#FAR -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#FAR
    ">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#
        DestinationDistance"/>
    <fzl:fmidHigh rdf:datatype="&xsd;double">100</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">100</fzl:fmax>
    <fzl:fmin rdf:datatype="&xsd;double">25</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">30</fzl:fmidLow>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#FLUID -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    FLUID">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#Traffic
        "/>
    <fzl:fmidLow rdf:datatype="&xsd;double">0</fzl:fmidLow>
    <fzl:fmin rdf:datatype="&xsd;double">0</fzl:fmin>
    <fzl:fmidHigh rdf:datatype="&xsd;double">33</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">35</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#JAM -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#JAM
    ">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#Traffic
        "/>
    <fzl:fmax rdf:datatype="&xsd;double">100</fzl:fmax>
```

```
    <fzl:fmidHigh rdf:datatype="&xsd;double">100</fzl:fmidHigh>
    <fzl:fmin rdf:datatype="&xsd;double">65</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">67</fzl:fmidLow>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#LOCAL -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    LOCAL">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#
        DestinationDistance"/>
    <fzl:fmin rdf:datatype="&xsd;double">10</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">15</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">15</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">30</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#LUNCH_TIME -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    LUNCH_TIME">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
        "/>
    <fzl:fmin rdf:datatype="&xsd;double">11</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">12</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">13</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">14</fzl:fmax>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#MORNING -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    MORNING">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
        "/>
    <fzl:fmidHigh rdf:datatype="&xsd;double">10</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">12</fzl:fmax>
    <fzl:fmin rdf:datatype="&xsd;double">6</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">8</fzl:fmidLow>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veem#NEAR -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
    NEAR">
    <rdf:type rdf:resource="http://ontology.bmw.de/veem#
        DestinationDistance"/>
    <fzl:fmin rdf:datatype="&xsd;double">0</fzl:fmin>
    <fzl:fmidLow rdf:datatype="&xsd;double">0</fzl:fmidLow>
    <fzl:fmidHigh rdf:datatype="&xsd;double">10</fzl:fmidHigh>
    <fzl:fmax rdf:datatype="&xsd;double">15</fzl:fmax>
```

```
    </owl:NamedIndividual>



    <!-- http://ontology.bmw.de/veem#NIGHT_EVENING -->

    <owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
        NIGHT_EVENING">
        <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
            "/>
        <fzl:fmin rdf:datatype="&xsd;double">22</fzl:fmin>
        <fzl:fmidLow rdf:datatype="&xsd;double">23</fzl:fmidLow>
        <fzl:fmax rdf:datatype="&xsd;double">24</fzl:fmax>
        <fzl:fmidHigh rdf:datatype="&xsd;double">24</fzl:fmidHigh>
    </owl:NamedIndividual>


    <owl:NamedIndividual rdf:about="http://ontology.bmw.de/veem#
        NIGHT_MORNING">
        <rdf:type rdf:resource="http://ontology.bmw.de/veem#DayTime
            "/>
        <fzl:fmin rdf:datatype="&xsd;double">0</fzl:fmin>
        <fzl:fmidLow rdf:datatype="&xsd;double">0</fzl:fmidLow>
        <fzl:fmidHigh rdf:datatype="&xsd;double">5</fzl:fmidHigh>
        <fzl:fmax rdf:datatype="&xsd;double">7</fzl:fmax>
    </owl:NamedIndividual>
</rdf:RDF>
```

Listing 7: A simple ontology model for infotainment

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY KIDS_ "http://ontology.bmw.de/veinfm#KIDS_&amp;amp;" >
    <!ENTITY NEWS_ "http://ontology.bmw.de/veinfm#NEWS_&amp;amp;" >
    <!ENTITY GAMES_ "http://ontology.bmw.de/veinfm#GAMES_&amp;amp;"
         >
    <!ENTITY SOCIETY_ "http://ontology.bmw.de/veinfm#SOCIETY_&amp;
        amp;" >
    <!ENTITY SCIENCE_ "http://ontology.bmw.de/veinfm#SCIENCE_&amp;
        amp;" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY RELIGION_ "http://ontology.bmw.de/veinfm#RELIGION_&amp
        ;amp;" >
    <!ENTITY GOVERNMENTS_ "http://ontology.bmw.de/veinfm#
        GOVERNMENTS_&amp;amp;" >
]>


<rdf:RDF xmlns="http://ontology.bmw.de/veinfm#"
     xml:base="http://ontology.bmw.de/veinfm"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:NEWS_="&NEWS_;amp;"
     xmlns:KIDS_="&KIDS_;amp;"
```

```
 xmlns:RELIGION_="&RELIGION_;amp;"
 xmlns:SOCIETY_="&SOCIETY_;amp;"
 xmlns:GAMES_="&GAMES_;amp;"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:SCIENCE_="&SCIENCE_;amp;"
 xmlns:GOVERNMENTS_="&GOVERNMENTS_;amp;">
<owl:Ontology rdf:about="http://ontology.bmw.de/veinfm">
    <dc:title>Vehicle Infotainment Model</dc:title>
    <dc:creator>St&#233;phane Turlier</dc:creator>
    <dc:language>en</dc:language>
    <rdfs:comment>limited version</rdfs:comment>
</owl:Ontology>


<owl:AnnotationProperty rdf:about="&dc;creator"/>
<owl:AnnotationProperty rdf:about="&dc;language"/>
<owl:AnnotationProperty rdf:about="&rdfs;comment"/>
<owl:AnnotationProperty rdf:about="&dc;title"/>


<!-- http://ontology.bmw.de/veinfm#hasParentTopic -->

<owl:ObjectProperty rdf:about="http://ontology.bmw.de/veinfm#
     hasParentTopic">
    <rdfs:domain rdf:resource="http://ontology.bmw.de/veinfm#
         EditorialTopic"/>
    <rdfs:range rdf:resource="http://ontology.bmw.de/veinfm#
         EditorialTopic"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>



<!-- http://www.w3.org/2002/07/owl#topObjectProperty -->

<owl:ObjectProperty rdf:about="&owl;topObjectProperty"/>


<!-- http://ontology.bmw.de/veinfm#InfotainmentDataProperty -->

<owl:DatatypeProperty rdf:about="http://ontology.bmw.de/veinfm#
     InfotainmentDataProperty"/>



<!-- http://ontology.bmw.de/veinfm#OmniphoneID -->

<owl:DatatypeProperty rdf:about="http://ontology.bmw.de/veinfm#
     OmniphoneID">
    <rdfs:domain rdf:resource="http://ontology.bmw.de/veinfm#
         Genre"/>
    <rdfs:subPropertyOf rdf:resource="http://ontology.bmw.de/
         veinfm#InfotainmentDataProperty"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

```
<!-- http://ontology.bmw.de/veinfm#RhapsodyID -->

<owl:DatatypeProperty rdf:about="http://ontology.bmw.de/veinfm#
    RhapsodyID">
    <rdfs:domain rdf:resource="http://ontology.bmw.de/veinfm#
        Genre"/>
    <rdfs:subPropertyOf rdf:resource="http://ontology.bmw.de/
        veinfm#InfotainmentDataProperty"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>


<!-- http://ontology.bmw.de/veinfm#EditorialFormat -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    EditorialFormat">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InformationDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#EditorialTopic -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    EditorialTopic">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InformationDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#EducationDescription -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    EducationDescription">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InformationDescription"/>
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InfotainmentDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#Genre -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#Genre">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#MusicDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#InformationDescription -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    InformationDescription">
```

```
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InfotainmentDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#InfotainmentDescription -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    InfotainmentDescription">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#Language -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#Language">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#EducationDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#Mood -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#Mood">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#MusicDescription"/>
</owl:Class>



<!-- http://ontology.bmw.de/veinfm#MusicDescription -->

<owl:Class rdf:about="http://ontology.bmw.de/veinfm#
    MusicDescription">
    <rdfs:subClassOf rdf:resource="http://ontology.bmw.de/
        veinfm#InfotainmentDescription"/>
</owl:Class>



<!-- http://www.w3.org/2002/07/owl#Thing -->

<owl:Class rdf:about="&owl;Thing"/>


<!-- http://ontology.bmw.de/veinfm#ARTS -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    ARTS">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#
        EditorialTopic"/>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#BLUES -->
```

```
<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    BLUES">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">469</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#BUSINESS -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    BUSINESS">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#
        EditorialTopic"/>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#CLASSICAL -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    CLASSICAL">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">399</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#COUNTRY -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    COUNTRY">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">196</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#DANCE -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    DANCE">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">267</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#FOLK -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    FOLK">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
```

```
    <RhapsodyID rdf:datatype="&xsd;string">490</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#HEALTH -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    HEALTH">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#
        EditorialTopic"/>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#JAZZ -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    JAZZ">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">229</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#LATIN -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    LATIN">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">305</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#NEW_AGE -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    NEW_AGE">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">436</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#OLDIES -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    OLDIES">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">424</RhapsodyID>
</owl:NamedIndividual>
```

```
<!-- http://ontology.bmw.de/veinfm#POP -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    POP">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">23</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#PUNK -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    PUNK">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">56</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#RAP -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    RAP">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">133</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#REGGAE -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    REGGAE">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">324</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#ROCK -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    ROCK">
    <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
        "/>
    <RhapsodyID rdf:datatype="&xsd;string">2</RhapsodyID>
</owl:NamedIndividual>



<!-- http://ontology.bmw.de/veinfm#SOUL -->

<owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
    SOUL">
```

```
        <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#Genre
            "/>
        <RhapsodyID rdf:datatype="&xsd;string">183</RhapsodyID>
    </owl:NamedIndividual>



    <!-- http://ontology.bmw.de/veinfm#TECHNOLOGY -->

    <owl:NamedIndividual rdf:about="http://ontology.bmw.de/veinfm#
        TECHNOLOGY">
        <rdf:type rdf:resource="http://ontology.bmw.de/veinfm#
            EditorialTopic"/>
    </owl:NamedIndividual>
</rdf:RDF>
```

Listing 8: Example of a user preference model (with persistent preferences)

```
<?xml version="1.0" encoding="MacRoman"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.0="http://ontology.bmw.de/veinfm#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:j.2="http://ontology.bmw.de/veem#"
    xmlns:j.1="http://ontology.bmw.de/fzl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:j.3="http://ontology.bmw.de/udpm#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://ontology.bmw.de/udpm#
      UserAPreference11">
    <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
        udpm#IMPORTANT"/>
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.8</j.3:fuzzyProperty>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
        LATIN"/>
    <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
        FLUID"/>
    <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
        FuzzyVariable"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
        Resource"/>
    <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.8</j.3:preferencePriority>
    <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
        UserPreference"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
        NamedIndividual"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ontology.bmw.de/udpm#
      UserAPreference10">
    <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.5</j.3:preferencePriority>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
        FuzzyVariable"/>
```

```
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      ROCK"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.5</j.3:fuzzyProperty>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      FLUID"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference1">
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      JAZZ"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      EVENING"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:fuzzyProperty>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:preferencePriority>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#IMPORTANT
    ">
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">60.0</j.3:fuzzyProperty>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">40.0</j.3:fuzzyProperty>
  <j.1:fmin rdf:datatype="http://www.w3.org/2001/XMLSchema#double
      ">40.0</j.1:fmin>
  <j.1:fmidLow rdf:datatype="http://www.w3.org/2001/XMLSchema#
      double">60.0</j.1:fmidLow>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <j.1:fmax rdf:datatype="http://www.w3.org/2001/XMLSchema#double
      ">100.0</j.1:fmax>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      InfotainmentPreference"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">100.0</j.3:fuzzyProperty>
  <j.1:fmidHigh rdf:datatype="http://www.w3.org/2001/XMLSchema#
      double">100.0</j.1:fmidHigh>
```

```
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
        Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ontology.bmw.de/udpm#NEGATIV">
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">-50.0</j.3:fuzzyProperty>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
        NamedIndividual"/>
    <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
        FuzzyVariable"/>
    <j.1:fmax rdf:datatype="http://www.w3.org/2001/XMLSchema#double
        ">0.0</j.1:fmax>
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.0</j.3:fuzzyProperty>
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">-25.0</j.3:fuzzyProperty>
    <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
        InfotainmentPreference"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
        Resource"/>
    <j.1:fmidLow rdf:datatype="http://www.w3.org/2001/XMLSchema#
        double">-50.0</j.1:fmidLow>
    <j.1:fmidHigh rdf:datatype="http://www.w3.org/2001/XMLSchema#
        double">-25.0</j.1:fmidHigh>
    <j.1:fmin rdf:datatype="http://www.w3.org/2001/XMLSchema#double
        ">-50.0</j.1:fmin>
  </rdf:Description>
  <rdf:Description rdf:about="http://ontology.bmw.de/udpm#
      UserAPreference2">
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.9</j.3:fuzzyProperty>
    <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
        FuzzyVariable"/>
    <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">0.9</j.3:preferencePriority>
    <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
        udpm#IMPORTANT"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
        Resource"/>
    <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
        LOCAL"/>
    <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
        POP"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
        UserPreference"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
        NamedIndividual"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ontology.bmw.de/udpm#SECONDARY
      ">
    <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
        FuzzyVariable"/>
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">60.0</j.3:fuzzyProperty>
    <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
        InfotainmentPreference"/>
    <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
        XMLSchema#double">40.0</j.3:fuzzyProperty>
```

```
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">-20.0</j.3:fuzzyProperty>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">20.0</j.3:fuzzyProperty>
  <j.1:fmidHigh rdf:datatype="http://www.w3.org/2001/XMLSchema#
      double">40.0</j.1:fmidHigh>
  <j.1:fmin rdf:datatype="http://www.w3.org/2001/XMLSchema#double
      ">-20.0</j.1:fmin>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.1:fmidLow rdf:datatype="http://www.w3.org/2001/XMLSchema#
      double">20.0</j.1:fmidLow>
  <j.1:fmax rdf:datatype="http://www.w3.org/2001/XMLSchema#double
      ">60.0</j.1:fmax>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserBPreference">
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      FAR"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.7</j.3:preferencePriority>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#SECONDARY"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.7</j.3:fuzzyProperty>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      CLASSICAL"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference3">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:fuzzyProperty>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      CONGESTIONNED"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:preferencePriority>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      ROCK"/>
```

```
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
     UserAPreference4">
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">0.7</j.3:fuzzyProperty>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">0.7</j.3:preferencePriority>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
       CLASSICAL"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
       FuzzyVariable"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
       udpm#IMPORTANT"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
       Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
       NamedIndividual"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
       NIGHT_MORNING"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
       UserPreference"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
     UserAPreference12">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
       Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
       NamedIndividual"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
       udpm#NEGATIV"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
       FuzzyVariable"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
       CLASSICAL"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">0.4</j.3:fuzzyProperty>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
       FLUID"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">0.4</j.3:preferencePriority>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
     UserAPreference5">
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
       RAP"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
       UserPreference"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
       FuzzyVariable"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
       NamedIndividual"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">1.0</j.3:preferencePriority>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
       udpm#NEGATIV"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
       XMLSchema#double">1.0</j.3:fuzzyProperty>
```

```
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      CONGESTIONNED"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAFARNEW_AGE">
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.0</j.3:preferencePriority>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.0</j.3:fuzzyProperty>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#SECONDARY"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      FAR"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      NEW_AGE"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#UserB">
  <j.3:BMWid rdf:datatype="http://www.w3.org/2001/XMLSchema#
      string">bmwid_2</j.3:BMWid>
  <j.3:userProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#string">bmwid_2</j.3:userProperty>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#User"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference6">
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      POP"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.7</j.3:fuzzyProperty>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      MORNING"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.7</j.3:preferencePriority>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
```

```
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference7">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:fuzzyProperty>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      ROCK"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      MORNING"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.8</j.3:preferencePriority>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/veem#
    NIGHT_EVENING">
  <rdf:type rdf:resource="http://ontology.bmw.de/veem#Context"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference8">
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.9</j.3:fuzzyProperty>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#NEGATIV"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      CLASSICAL"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.9</j.3:preferencePriority>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      MORNING"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/udpm#
    UserAPreference9">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <j.3:preferencePriority rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.9</j.3:preferencePriority>
  <rdf:type rdf:resource="http://ontology.bmw.de/fzl#
      FuzzyVariable"/>
```

```
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://ontology.bmw.de/udpm#
      UserPreference"/>
  <j.3:hasConsequent rdf:resource="http://ontology.bmw.de/veinfm#
      POP"/>
  <j.3:hasConsequentValue rdf:resource="http://ontology.bmw.de/
      udpm#IMPORTANT"/>
  <j.3:hasAntecedent rdf:resource="http://ontology.bmw.de/veem#
      AFTERNOON"/>
  <j.3:fuzzyProperty rdf:datatype="http://www.w3.org/2001/
      XMLSchema#double">0.9</j.3:fuzzyProperty>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://ontology.bmw.de/veem#AFTERNOON
    ">
  <rdf:type rdf:resource="http://ontology.bmw.de/veem#Context"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      NamedIndividual"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/2000/01/rdf-schema#
    Container">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Container"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#object">
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-
      syntax-ns#Statement"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax
      -ns#Property"/>
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/1999/02/22-
      rdf-syntax-ns#object"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#Seq">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Container"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Resource"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
      Resource"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf
      -syntax-ns#Seq"/>
</rdf:Description>
```

Listing 9: Simple ontologic model for fuzzy variables

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY fzl "http://ontology.bmw.de/fzl#" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY owl11 "http://www.w3.org/2006/12/owl11#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl11xml "http://www.w3.org/2006/12/owl11-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<rdf:RDF xmlns="http://ontology.bmw.de/fzl#"
     xml:base="http://ontology.bmw.de/fzl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl11="http://www.w3.org/2006/12/owl11#"
     xmlns:owl11xml="http://www.w3.org/2006/12/owl11-xml#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:fzl="http://ontology.bmw.de/fzl#">

    <owl:Ontology rdf:about="http://ontology.bmw.de/fzl">
        <dc:language>en</dc:language>
        <rdfs:comment>limited version</rdfs:comment>
        <dc:title>Fuzzy Logic Ontology</dc:title>
        <dc:creator>St&#233;phane Turlier</dc:creator>
    </owl:Ontology>


    <!-- http://ontology.bmw.de/fzl#fmax -->

    <owl:DatatypeProperty rdf:about="#fmax">
        <rdfs:range rdf:resource="&xsd;double"/>
        <rdfs:domain rdf:resource="#FuzzyVariable"/>
    </owl:DatatypeProperty>



    <!-- http://ontology.bmw.de/fzl#fmidHigh -->

    <owl:DatatypeProperty rdf:about="#fmidHigh">
        <rdfs:range rdf:resource="&xsd;double"/>
        <rdfs:domain rdf:resource="#FuzzyVariable"/>
    </owl:DatatypeProperty>



    <!-- http://ontology.bmw.de/fzl#fmidLow -->

    <owl:DatatypeProperty rdf:about="#fmidLow">
        <rdfs:domain rdf:resource="#FuzzyVariable"/>
        <rdfs:range rdf:resource="&xsd;double"/>
    </owl:DatatypeProperty>
```

```
    <!-- http://ontology.bmw.de/fzl#fmin -->

    <owl:DatatypeProperty rdf:about="#fmin">
        <rdfs:range rdf:resource="&xsd;double"/>
        <rdfs:domain rdf:resource="#FuzzyVariable"/>
    </owl:DatatypeProperty>

    <!-- http://ontology.bmw.de/fzl#FuzzyVariable -->

    <owl:Class rdf:about="#FuzzyVariable"/>
</rdf:RDF>
```

La mise en page de ce manuscrit reprend le modèle pour LaTeX *Classic Thesis* créé par André Miede.

Peut-on écouter, quand on le veut, le tube de l'été sur la route des vacances sans passer chez un disquaire ? Comment créer un programme radio personnalisé avec musique, informations et divertissement pour nous accompagner dans les embouteillages ? Est-il possible de découvrir de nouveaux horizons musicaux tout en conduisant ?

L'internet mobile pour les véhicules devrait permettre bientôt de répondre à ces questions encore faut-il s'assurer que les voitures soient capables d'accompagner le changement que nous proposent les médias à la demande. Cette thèse étudie les adaptations, aménagements et nouveaux composants qu'il faut apporter à la télématique automobile pour proposer aux conducteurs et à leurs passagers une nouvelle expérience multimédia dans les véhicules.

Tout d'abord, l'architecture de distribution des médias doit être aménagée afin de permettre une abstraction des très nombreux fournisseurs de contenu, et de simplifier le client multimédia embarqué dans l'autoradio. Par ailleurs, les applications internet ayant tendance à nécessiter une grande interactivité, il faut optimiser l'usage des modalités à disposition du conducteur afin de ne pas le dévier de sa tâche principale : la conduite. Enfin, on peut pousser l'automatisation de la personnalisation jusqu'à utiliser des informations contextuelles pour créer des listes de lectures qui changent suivant l'heure de la journée, les conditions de circulation ou encore la distance à parcourir.

Réalisée au sein du département de recherche en architectures logicielles embarquées de *BMW Group Research and Technology* à Munich, la thèse est illustrée par des exemples tirés des prototypes qui ont été développés pour le constructeur bavarois afin d'explorer les nombreuses possibilités offertes par les médias connectés.