CONCERT: A New Framework for Contextual Computing in Tourism to Support Human Mobility

Submitted to the School of Engineering at the University of Deusto in partial fulfillment of the requirements for the degree of Doctor of Philosophy

May 20, 2010

By:
Carlos Lamsfus Franco

Supervisors:
Aurkene Alzua-Sorzabal, Ph.D.
and
Dr. Diego López de Ipiña González de Artaza

Abstract in English

Human mobility today is central in societies and its comprehension is essential to understand the current way of life. Recently, intensive scientific efforts have been invested in better understanding this phenomenon. eTourism represents a very active research field in Computer Science. In particular, in the realm of ubiquitous computing. The evolution of mobile devices and their proliferation in society, the advancement of communication technologies and the trend toward creating hybrid spaces (symbiosis between nature and technology) will trigger a radical change in the way persons *en route*, "visitors" in operative terms, interact with their environment.

In the near future most of these people will have spent a large part of their (adult) lives in the digital era. Time will continue to be an increasingly scarce resource, and occasional visitors of this kind will demand anytime, anywhere kind of services adapted to both their personal and environmental contexts. Therefore, context-based applications will not only shape the future of tourism services, but also provide with the opportunity to better understand human behaviour in the future knowledge society.

Considering this background, the work developed within this dissertation suggests a new approach to the theory of context. Starting from an analysis of the intrinsic characteristics of human mobility, the objective is to work on a model that meets the contextual needs and requirements of visitors. Thus, the difficulty lies in determining what information is necessary to define visitors' context: which is the minimum amount of that information required, where that information is and how it can be effectively and practically retrieved to be subsequently processed.

To tackle this problem a new definition of the notion of context is proposed. The new conception implies new context requirements in terms of information and sources of information necessary. This way, populating whole areas, cities or regions with sensors would just not be affordable. Therefore, the approach this works takes, proposes to use other sources of (contextual) information that are hosted in digital environments, i.e. in the Internet. These new sources of information do not require specific and

expensive infrastructures from an economic and implementation points of view. The information gathered from the Internet will then be completed with data coming from the mobile device embedded location sensors and with information provided by visitors themselves.

In addition, this work suggests to experiment with a particular communication technology, which is not new but has never been used for disseminating neither contextual nor tourism information: digital broadcasting. The information received by mobile devices will be processed according to semantic-based rules implemented on top of a network of ontologies. The rules filter incoming information, they classify it and then only display the information that is really relevant for visitors given their current context, i.e. according to the specific context values, that have been established by the new definition of context.

Thus, some of the restrictions of current approaches to context-awareness are overcome: Firstly, the model suggested focuses on visitors instead of on particular systems running in specific environments. Secondly, visitors are modelled according to various parameters that have been validated by the tourism scientific community. And thirdly, sensors do not restrict the use of such contextual computing services to areas that had been previously populated with them. In addition, the experiment performed with digital broadcasting allows to acquire context information anywhere, anytime, i.e. anywhere visitors may be located at a given moment, following the stipulations of the ubiquitous computing paradigm.

The theoretical -the novel approach to find simple structures and model the context of visitors *en route*-, practical and experimental results obtained from the implementation of the developed ontology in feasible scenarios suggest significant impact in the realm of contextual computing on general and on contextual computing in tourism in particular.

Abstract in German

Menschliche Mobilität ist heute in den Gesellschaften ein zentrales Thema und sein Verständnis ist wesentlich, die gegenwärtige Lebensart erfassen zu können. In der letzte Zeit sind intensive wissenschaftliche Bemühungen investiert worden, um dieses Phänomen besser zu verstehen. Die wissenschaftlichen Aspekte des eTourismus stellen heutzutage ein wichtiges Forschungsgebiet der angewandten Informatik dar, insbesondere im Bereich des *Ubiquitous Computing*. Die Entwicklung mobiler Geräte und ihre breite Anwendung in der Gesellschaft, die Durchdringung der Kommunikationstechnologien in Verbindung mit der Tendenz zur Schaffung immer neuer hybrider Räume als einer Symbiose zwischen Natur und Technik werden radikale Veränderungen in der Art und Weise mit sich bringen, in der der mobile Mensch – etwa der Besucher eines Ortes – mit seiner Umgebung interagiert.

Schon in naher Zukunft werden die meisten Erwachsenen einen großen Teil ihres Lebens in der digitalen Ära verbracht haben. Die Zeit wird auch zukünftig eine in immer weiter zunehmendem Maße eine knappe Ressource sein, und der mobile Mensch wird überall und jederzeit Dienstleistungen verlangen, die seinen persönlichen Bedürfnissen und Umgebungskontexten entsprechen. Solche kontextbasierten Anwendungen werden nicht nur die Zukunft der Tourismusdienstleistungen verändern, sondern bieten gleichzeitig eine Gelegenheit, menschliches Verhalten in der Wissensgesellschaft der Zukunft besser zu verstehen und vorherzusagen.

Vor diesem Hintergrund wird in der vorliegenden Dissertation ein neuer Ansatz zur Theorie des Kontextes entwickelt. Basierend auf einer Analyse der Eigenschaften der menschlichen Mobilität, besteht ein Ziel darin, ein Modell zu erarbeiten, das den Ansprüchen und Anforderungen der Menschen bezüglich des Kontextes genügt. Auf diese Weise liegt die Schwierigkeit darin festzustellen, welche Informationen für die Definition des Kontextes notwendig sind: woraus die mindestens notwendigen Informationen bestehen, wo diese vorliegen und wie sie effizient abgerufen werden können, um nachher verarbeitet zu werden.

Um dieses Problem anzugehen, wird eine neue Definition des Begriffes Kontext vorgeschlagen. Dieses neue Konzept stellt neue Anforderungen an die Art der Informationen und die nötigen Informationenquellen. Da die Ausstattung von Städten oder ganzen Regionen mit Sensoren ökonomisch untragbar wäre, auch auf Implementierungsgrunde, liegt der Ansatz der vorliegenden Arbeit darin, andere Quellen kontextueller Informationen zu verwenden, die im digitalen Umfeld bzw. im Internet gefunden werden können.

Diese neuen Informationsquellen erfordern keine spezifischen und kostspieligen Infrastrukturen in Bezug auf die Kosten und den Aufwand der Implementierung. Die im Internet gefundenen Informationen werden mit Daten ergänzt, die von in mobilen Geräten eingebetteten Positionssensoren stammen, und mit Angaben, die vom Benutzer selbst zur Verfügung gestellt werden.

Zusätzlich wird in der vorliegenden Arbeit mit einer bewährten Kommunikationstechnologie experimentiert, die zwar nicht neu ist, die aber nie weder im Bereich Kontext noch zur Verbreitung von Tourismusinformationen benutzt wurde: das digitale *Broadcasting*. Die von den mobilen Geräten erhaltenen Informationen werden gemäß semantischen Regeln weiterverarbeitet, die auf einem Netz von Ontologien basieren. Die Regeln filtern ankommende Informationen, klassifizieren sie und zeigen dann nur die Informationen, die für die Besucher unter Berücksichtigung des jeweiligen Kontextes wirklich relevant sind, d. h. unter Verwendung konkreter Kontextwerte gemäß der neuen Definition.

Damit werden einige Einschränkungen vieler kontextbasierter Systeme überwunden: Erstens, zielt das vorgeschlagene Modell direkt auf den Besucher ab, anstatt indirekt auf Systeme, die nur extern mit dem Benutzer zusammenhängen. Zweitens werden Besucher durch Verwendung verschiedener Parameter modelliert, die von der wissenschaftlichen Gemeinschaft im Tourismus validiert worden sind. Drittens schränken verwendete Sensoren den Gebrauch solcher kontextbasierter Anwendungen nicht nur auf bestimmte Bereiche ein, die vorher mit ihnen ausgestattet wurden. Das durchgeführte Experiment unter Verwendung von digitalem *Broadcasting* zeigt, dass Kontextinformationen dem Benutzer jederzeit und überall zur Verfügung stehen und entsprechen damit dem Paradigma des *Ubiquitous Computing*.

Die theoretischen (die innovative Annäherung einfache Strukturen zu finden, der Kontext des Besuchers modellieren zu können), praktischen und experimentellen Ergebnisse dieser Forschungsarbeit sind damit allgemein auf dem Gebiet des *Contextual Computing* im Allgemeinen relevant, insbesondere aber im Anwendungsbereich des Tourismus.

Abstract in Spanish

La movilidad humana representa una cuestión central en la Sociedad y su comprensión es esencial para entender la forma de vida actual. La comunidad científica ha invertido mucho esfuerzo recientemente para entender este fenómeno. La actividad científica en eTurismo constituye, en la actualidad, un campo de investigación muy activo en el ámbito de las ciencias de la computación. En especial, en la rama de la computación ubicua. La evolución de los dispositivos móviles y su alta permeabilidad social, de las tecnologías de la información y comunicación y la tendencia a generar espacios híbridos (cercanos a una simbiosis entre naturaleza y tecnología) provocarán un cambio radical en la manera en que las personas *en ruta*, visitantes en términos operativos, interactuarán con su entorno.

En un futuro próximo, la mayoría de las personas habrán pasado gran parte de sus vidas (adultas) en la era digital. El tiempo continuará siendo un recurso (cada vez más) escaso y este tipo de visitantes ocasionales demandará servicios en cualquier lugar, en cualquier momento, adecuados a su contexto personal y ambiental. Por lo tanto, las aplicaciones basadas en el contexto representan no sólo el futuro de los servicios turísticos, sino también la oportunidad para comprender mejor el comportamiento humano en la sociedad del conocimiento del futuro.

Con este marco de fondo, el trabajo desarrollado en esta tesis propone una nueva aproximación al estudio de la teoría de contexto. Partiendo de un análisis de las características intrínsecas de la naturaleza de la movilidad humana, el objetivo es desarrollar un modelo que ofrezca respuestas eficaces a los retos derivados de las necesidades contextuales de un visitante como tal. Así, la dificultad reside en determinar qué información es necesaria para definir el contexto de un visitante, cuál es la información mínima necesaria para definir dicho contexto, dónde se encuentra y cómo se puede extraer de manera efectiva y práctica para ser, posteriormente, procesada.

Para abordar el problema descrito se propone una nueva definición del concepto de contexto. La nueva concepción implica nuevos requerimientos en términos de información necesaria y de fuentes de información de contexto.

De esta manera, instalar sensores en ciudades o regiones enteras no sería viable. Por ello, la aproximación propuesta en este trabajo propone utilizar otras fuentes de información (contextuales) que residen ya en el entorno digital, i.e. Internet y que no requieren el desarrollo de infraestructuras específicas y costosas, tanto desde el punto de vista económico como de implementación. La información recogida de Internet se complementará con datos provenientes de los sensores embebidos en los dispositivos móviles y proporcionada por los propios visitantes. El modelo de contexto sitúa al visitante en su centro y lo caracteriza de acuerdo a parámetros estándares definidos y consensuados por la comunidad científica en turismo.

Además, este trabajo propone experimentar con una tecnología de comunicación que, sin ser nueva, no se ha utilizado para difundir información turística y de contexto anteriormente: la radio digital. La información recibida por este medio en los dispositivos móviles será procesada por medio de reglas basadas en una red de ontologías. Las reglas filtran la información recibida, la clasifican y muestran, únicamente, la que es de verdad relevante para los visitantes en función de su contexto, es decir, en función de los valores concretos (establecidos por la nueva definición) que caracterizan su contexto, i.e. la situación de un visitante.

De esta manera, se corrigen algunas de las restricciones que evitan la adopción masiva de sistemas basados en contexto: en primer lugar, el modelo propuesto se centra en el visitante en lugar de en un sistema externo a la persona. En segundo lugar, los visitantes se modelan con arreglo a una serie de parámetros establecidos y aceptados. En tercer y último lugar, los sensores no obligan a utilizar las aplicaciones que pudieran derivarse de este marco teórico en ubicaciones concretas. El experimento realizado con la radio digital permite adquirir la información de contexto en cualquier momento y en cualquier lugar, es decir, donde sea que se encuentre un visitante y siguiendo los requerimientos establecidos en el paradigma de la computación ubicua.

Los resultados teóricos (nueva aproximación al contexto para hallar estructuras simples y modelar el contexto de los visitantes *en ruta*), prácticos y experimentales de esta investigación sugieren que el trabajo desarrollado puede tener un impacto significativo en el ámbito de la computación contextual en general y en su aplicación en el turismo en particular.

Abstract in Basque

Giza mugikortasuna gai zentrala da gizarteetan eta bere ezagutza ezin bestekoa da gaur egungo bizimodua ulertu ahal izateko. Komunidade zientifikoak ahalegin haundiak egin ditu gai hau hobe ulertzen saiatzeko. eTurismoaren inguruko jarduera zientifikoa oso ikerketa lerro aktiboa da gaur egun konputazio zientzietan, bereziki nonahiko konputazioan. Pertsonek mugikortasunean -bisitariak alegia- inguruguneareki elkar eragiteko duten modua errotik aldatuko da hainbat aldagai direla eta: gailu mugikorren nahiz informazio eta komunikazio teknologien garapen etengabea; gizartean hedatzeko duten ahalmen handia eta, guzti honen ondorioz, eremu hibridoak sortzeko joera, natura eta teknologiaren arteko sinbiosi batera hurbilduz.

Etorkizun hurbilean, jende gehienaren bizialdiaren zati nagusia aro digitalean pasatakoa izango da. Denborak baliabide geroz eta urriagoa izaten jarraituko du eta halako noizbehinkako bisitariek zerbitzuak nonahi eta noiznahi eskatuko dituzte, haien kontexktu pertsonal eta ingurunekora egokituta gainera. Hortaz, kontextuan oinarritutako aplikazioak, turismo zerbitzuen geroa ezezik, etorkizunezko ezagutzan oinarritutako gizartean giza jokabidea hobe ulertzeko aukera ere badira.

Azaldutako esparru honetan oinarrituta, tesi honetan garatutako lanak kontextuaren teoriara beste hurbilpen bat proposatzen du. Giza mugikortasunaren berezko ezaugarrien analisi batetik abiatuta, helburua, bisitarien kontextu-beharrek sortzen dituzten erronkei erantzun eraginkorrak ematen dizkien eredu bat lantzea da. Honela, zailtasun nagusia bisitari baten kontextua definitzen duen informazioa zehaztea da: zein den kontextu hori definitzeko behar den gutxieneko informazioa, non dagoen informazio hori eta nola jaso daiteken modu praktiko eta eraginkorrean, ondoren eraldatu ahal izateko.

Arazo honi aurre egiteko, kontextu kontzeptuaren definizio berri bat proposatzen da. Kontzepzio berri honek eskakizun berriak dakartza berarekin, kontextuaren informazio motari eta informazio iturriei dagokienean. Gune, hiri edo eskualde osoak sensorez betetzea bideragarria ez denez, lan honek beste bide bat proposatzen du: ingurune

digitalean dagoeneko badauden beste informazio iturri batzu erabiltzea, kontextukoak direnak eta azpiegitura berezi eta garestiak (ekonomikoki nahiz inplementazioaren ikuspegitik garestiak) garatzen eskatzen ez dutenak, hau da: Internet erabiltzea proposatzen da. Internetetik jasotako informazioa, gailu mugikorran enbebituta dauden kokapen sensoreek eta bisitariak berak emandako informazioarekin osatuko da. Kontextu eredu berri honek, bisitaria erdigunean kokatzen du eta turismoaren arloko komunitate zientifikoak ezarritako parametroen arabera definitzen du.

Gainera, tesi honek saiakuntzak egitea proposatzen du, berria izan gabe, turismoaren eta kontextuko informazioaren arloetan sekula erabili ez den teknologia batekin: irrati digatalarekin. Teknologia honen bidez gailu mugikorrean jasotako informazioa, ontologi sare batean oinarrituta dauden arau batzuekin prozesatuko da. Arauok informazioa iragazi, sailkatu, eta bisitarientzat bere kontextuaren arabera esanguratsua dena bakarrik erakutsiko dute; alegia, bere kontextuak, i.e. egoerak, ezaugarri dituen balore zehatzen arabera -definizio berriak ezarritakoak- esanguratsu dena.

Horrela, sistema hauek orokorki erabili ahal izateko dauden muga batzu ekiditen dira: batetik, proposatutako ereduak bisitariarengan jartzen du arreta. Bestetik, bisitariak aurretik finkatutako parametru eta irizpide batzuen bidez modelizatzen dira. Azkenik, sensorerik erabiltzen ez denez, eredu teoriko honen aplikazioak ez dira mugatzen aurretik sensoreak jarri dituzten kokagune jakinetara: irrati digitalarekin egindako saiakuntzak aukera ematen du kontextuko informazioa edonoiz eta edonon biltzeko, bisitaria dagoen edozein tokitan, eta nonahiko konputazioaren paradigmaren printzipioak jarraituz.

Ikerketa lan honetan lorturiko emaitza teoriko (estruktura simpleak bilatzeko egindako hurbilketa berria eta bisitarien kontextu eredu berria), praktiko eta esperimentalak adierazten duten moduan, emaitza hauek eragin nabarmena eduki dezakete orokorrean kontextuko konputazioan eta bereziki turismoarekin lotutako kontextuko konputazioan.

Acknowledgements

"I can no other answer make, but, thanks, and thanks."
William Shakespeare, (1564 - 1616), English poet and playwright.

Kontestu hitza latinetik dator. Jatorriz contextus *edo* con-textere *"ehotu" edo "elkarrekin ehotu" esan nahi du. Tesi bat bakarkako lan bat den arren, orain nagoen momentura heldu ahal izateko nere atzetik jende askoren laguntza ehotu da elkarrekin, nik behar izan dudan kontestu aproposa sortzeko eta horrela, lan hau burutu ahal izateko.*

Lehenik eta behin, nere tesi zuzendariak, Aurkene Alzua-Sorzabal eta Diego López de Ipiña doktoreak eskertu nahi ditut emandako laguntza eta zuzendaritzarengatik. Batez ere, azkeneko hilabeteetan eduki ditugun elkarizketa sakon eta produktiboengatik. Zuei, mila esker. Bereziki, Aurkene eskertu nahiko nuke ere, ez bakarrik tesi hontan egin duen lanarengatik baizik eta orain lana egiten dugun ikerketa zentrua ideia bat besterik ez zenean nigan pentsatzeagatik ideia hori errealitate bihurtu zedin. Eskerrik asko.

Gertuen eduki ditudan lankideak ere ezin ditut ahaztu. Human Mobility and Technology *ikerketa lerroa osatzen duten lankide guztiak eskertu nahi ditut, oso bereziki David Martín. Gracias, David, por ser mi* wikipedia *caminante de Java y por desatascarme en los momentos más delicados del desarrollo. CICtourGUNEko gainontzeko lankideei ere bai eskerrak eman nahi nizkioke prozesu guztian zehar nere bizitza errezago egiteagatik. Denei, mila esker.*

Ich möchte mich auch bei Herrn Univ. Prof. Dipl.- Ing. Dr. techn. Hannes Werthner bedanken, der mich die Gelegenheit gegeben hat, einen Aufenthalt an der TU-Wien zu machen. Vielen Dank Prof. Wethner für Ihre Zeit und Kommentare, die beigetragen haben, um die Annäherung der Dissertation zu verbessern. In der gleichen Weise möchte ich mich bei der ganze eCommerce Gruppe danken, die mich Gefühl wie in Haus vom ersten Tag bildete. In besonders, ich will Mag. rer. soc. oec. Christoph Grün danken, für seine Kommentare über die Dissertation und auch, für seine

allgemeine Unterstützung während meiner Aufenthalt in Wien. An alle, vielen vielen dank für die unvergesslich Erfahrung.

I would like to thank Dr. Tim Smithers for having given me the passion for research and science. Thank you, Tim, for your support and friendship.

Me gustaría también agradecer a todos mis amigos, por seguir contando conmigo para hacer planes aunque supieran de antemano que la respuesta iba a ser "no". Gracias a Mikel, Marcos, Luis Alberto, Alberto, Saly, Carlos (Toro) -también por tus comentarios y reflexiones acerca de las ontologías-, Olatz -zure interes, pazientzia eta laguntzagatik-, Ernest for understanding my "unavailability" *y a todos los que habéis estado de alguna u otra forma involucrados en mi vida. A mis compañeros de VICOMTech, donde empecé mi carrera de investigación, por cambiarme la visión de las cosas y por darme la oportunidad de descubrir mi vocación.*

Thanks for the lovely music, Bruce...

Nere bi familiei eskerrak eman nahi dizkiet. Nere famili berriari, prozesu guztian zehar hor egoteagatik, zuen laguntzarengatik, Carlosi mahaiagatik(!), nere ez-egoteak ulertzeagatik eta zuen interesarengatik egiten egon naizen lanarengatik. Txikienei, Ekaitz eta Oihan, jakin gabe egin duzuen guztiarengatik: indarra eman, deskonektatzen lagundu eta irrifarre bat jarri momentu txarrenetan. Zuei denei, mila, mila esker.

Nere betiko familia ere eskertu nahi dut, batez ere, beraien ulermen eta laguntzarengatik. Bereziki, noski, nere gurasoak eskertu nahi ditut eta tesi hau dedikatu. Prozesu guztian animoak ematen egoteagatik, idatzitakoa irakurtzeagatik eta batez ere, bizitzan zehar besterekiko dedikazio adibide bat izateagatik. Zuei, bihotzez, mila esker. Ezin dut momentu hauetan nere izeba Ana ahaztu, gaixotasunaren aurrean superazio eta adore adibide bat izaten ari delako.

Azkenik, Karmele, nere emaztea, eskertu nahi dut. Bera bai izan delako egitazko context-aware *(giza) entidade bat eta bere kontestura edozein beste gailu baino hobe egokitzen jakin izan duelako. Beti jakin izan duzu zer esan, nola esan, zer egin, nola lagundu,..., zutaz ahaztu eta ni laguntzen egon zara denbora luze honetan. Mila esker, Karmele, zure laguntzagatik, zure eritzia emateagatik..., danagatik. Zuri ere tesi hau dedikatzen dizut.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *"Nothing in this world can take the place of persistence. Talent will not; nothing is more common than unsuccessful people with talent. Genius will not; unrewarded genius is almost a proverb. Education will not; the world is full of educated derelicts. Persistence and determination alone are omnipotent."*
> *Calvin Coolidge, (1872 - 1933), 30th President of the United States of America (1929 - 1933).*

The Travel and Tourism industries are some of the most significant industries worldwide. In fact, the two most representative bodies in the realm of tourism research, namely the United Nations World Tourism Organization (UNWTO)[1] and the World Travel and Tourism Council(WTTC)[2], reported 880 million international arrivals in 2009. The severe global financial and economic crisis affected tourism only in part, with a slight decrease (4%) in the number of international arrivals in 2008. Despite the final figures being a bit lower than initially forecasted by the UNWTO, they were still significantly relevant. In addition, the tourism industries as a whole generated over 230 million (direct) jobs and produced over 10% of the global Gross Development Product (GDP).

Tourists are intensive technology users[GYF00] who increasingly use the Internet not only to gather information on their next destination, but also to check transportation facilities, accommodation opportunities, restaurants, events, museums and so forth[SF10]. They also consider information provided by other tourists (the Web 2.0 phenomena[3]) who have been at

[1] http://www.unwto.org

[2] http://www.wttc.org

[3] http://oreilly.com/web2/archive/what-is-web-20.html

the same destination[ICB09]. More importantly, tourists use the Internet to purchase tourism services[Wer03] [OM10]. In fact, according to the report published by Carl H. Marcussen, online travel sales increased by 17% from 2007 to 2008 and reached EUR 58.4 billion in the European market in 2008, i.e. 22.5% of the market. In 2009 a further increase of about 12% was expected to reach about 25% of the market, i.e. one in every four Euros spent online was devoted to travel and tourism products or services[4].

Mobile devices are also in continuous development. Their functionalities, communication features and computing capacities have grown exponentially [CK01][Har] in the last decade. Just as an example, mobile devices including the iPhone[5], Google's Android G1[6] or Nexus One[7], the Nokia E series[8] or devices such as the Netbooks are fast blurring the line between phones and mobile devices in terms of both power and functionality, to the extent that these mobile devices will eventually be the primary access devices to online services[SIT09].

Furthermore, (wireless) communication technologies, such as Third Generation mobile technology (3G), Universal Mobile Telecommunications Systems (UMTS), Wireless Fidelity (Wi-Fi) and Worldwide Interoperability for Microwave Access (WiMAX) have evolved rapidly[Inc07] [Kah05] [BP08] [CK01] enabling people to be continuously connected to online services in an almost anywhere anytime manner at a reasonable cost (when not subject to roaming services, where prices are as much as EUR 1 per megabyte -expected to be EUR 0.80 as of July 1st 2010 and EUR 0.50 as of July 1st 2011)[9] following the promises of ubiquitous computing[Wei99], from an information access point of view at least.

The mass adoption of these kinds of devices and technologies by everyday users in today's society is producing new behaviour patterns and profoundly changing general consumer habits[LAS09] [SRVN10]. In addition, information-intensive industries are developing new products and services that are based on the consumption of information[Poo03] [She97] [Buh03]. One of these industries, tourism, is undergoing a radical transformation

[4]Trends in European Internet Distribution of Travel and Tourism Services. Source: Centre for Regional and Tourism Research, Denmark. http://www.crt.dk

[5]http://www.apple.com/iphone

[6]http://www.androidg1.org/

[7]http://www.google.com/phone/?locale=en$_USs7e=$

[8]http://www.nokia.com

[9]Source: Europa Press release, http://europa.eu

from a labour-intensive industry to an information- and knowledge-intensive industry[Wer03] [BFP$^+$09] where information has become a primary asset.

However, the amount of information on the Internet is growing constantly at exponential rates (see Fig.1.1) [GHF06]. There are an estimate of at least 20.73 billion indexed Websites[10] in April 10^{th} 2010. This information overload creates (cognitive) problems to end-users and makes finding the right piece of information a real challenge. This can be even more difficult for people using mobile devices, given the small screen and keyboard, thus posing serious restrictions for human interaction in Web browsing. Therefore, given that mobility is the essence of tourism and that tourists are people on the move, who use mobile devices most of the times two questions arise: how can they retrieve the right piece of information and how can the right piece of information be (automatically) sent to them at a given moment of time?

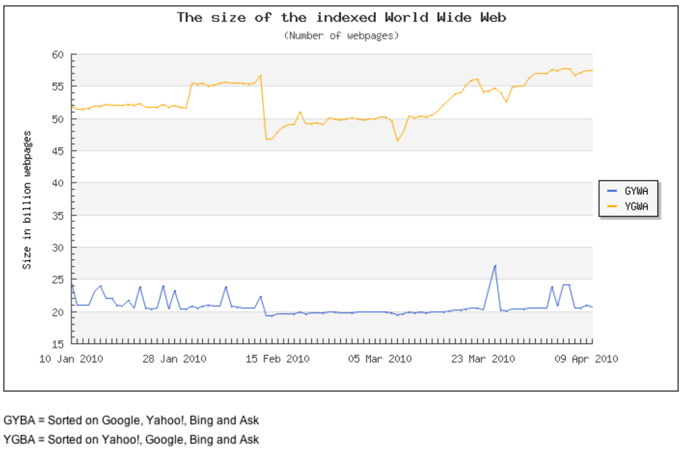

Figure 1.1: Estimate number of Websites in January 2010. Source: http://www.worldwidewebsize.com/

One possible way to attempt this would be by using **context information**. If there are means to help make a sound understanding of what context is, operatively and rigorously define it and its scope, determine the variables that constitute it and translate that conceptual notion into a computing model, then the model could be used to automatically filter, or select the information existing on the Web for people on the move, i.e.

[10]Source: http://www.worldwidewebsize.com/

visitors [UNWTO International Recommendations for Tourism Statistics, 2008, p.10]. In this sense, visitor context is crucial as it can be used as a kind of filter to access a particular piece of information that is relevant to support and enhance visitors' mobility.

Figure 1.2: Constituents of Context within the framework of Human Mobility

Some studies and research show that context-based applications represent an opportunity for future services within the travel and tourism industries [GW04][GPW$^+$08] and the possibility to better understand human behaviour in the future knowledge society. By 2015 there will be approximately 3 billion people moving around the world[SIT09] and regional and international visitor flows will be multiplied by three[LAS09]. Most of these people will have lived the majority of their (adult) lives in the digital era and their behavioural patterns with their environment will be significantly different from today's standards. Time will continue to be a very scarce resource and this new type of travellers will require anytime, anywhere, personalized kind of services, i.e. context-dependent or ubiquitous services[HSLF08].

The *leitmotiv* and the real **objective** of this dissertation is to **focus on contextual computing in the framework of human mobility**. This piece of research begins with a thorough review of the literature and other related research works. It analyses them and extracts a number of conclusions regarding the reasons for the lack of such context-aware applications for everyday users.

Then, based on the nature of human mobility and the restrictions encountered in the existing approaches, the dissertation proposes to create a new framework for context awareness, i.e. contextual computing in tourism. This new framework, named CONCERT, has three important differences with respect to traditional approaches: (i) it studies the notion of context from the perspective of Human Mobility; (ii) it does not require external sensors to gather contextual information, and (iii) it uses digital broadcasting as communication technology.

1.1 Problem description and motivation

Since Mark Weiser enunciated the vision of Ubiquitous Computing, a new interaction paradigm with computers[Wei99], intensive research has been conducted within the so-called context-awareness research discipline[Ay07] [VG07] . However, despite all the effort invested, these types of systems are not yet available to the general public and only exist in certain laboratory environments[LASM$^+$09a] [RPB10]. At the same time, location and context-based information systems still have to get off the ground in the field of tourism[BP08].

There are a number of reasons for this:

- There is no sufficient consensus on a conceptual framework for context-based applications[DA00a][GPZ94][Ay07] and thus, there is no operative definition of the notion of context.
- As a consequence, it has not yet been possible to establish neither a common or standard framework, a methodology nor tools to model context and manage context information[Ay07][GPZ94].
- Context has never been studied as such, but as a tool for other research disciplines such as, human-computer interaction[Dey00], intelligent software agents[CF03] or distributed systems[SLP04]. These works show an oversimplified approach to context and authors use contextual information to enhance their systems' functionalities but not for the sake of studying context itself.
- The notion of context has been approached regarding the environment in which a system runs and only indirectly tackling the main characteristics of humans, making the resulting applications necessarily very much techno-centric.

- Existing context-aware applications have been developed in an *ad-hoc* manner and need a number of pre-requisites in order to work properly.

- Most context-aware applications require populating areas of interest with networks of sensors to gather contextual information. This highly increases the complexity of context-aware systems by adding hardware difficulties derived from the use of sensors. It also greatly restricts the use of context-aware applications to the (particular) areas populated with network of sensors that are compliant to the application.

Around 90% of the visitors around the World carry a mobile device with them at all times and require in some way or another to be connected to sources of information such as the Internet[SIT09]. Context-based applications are the opportunity and the future in the travel and tourism industry[BCGB07]. The unexploited potential of contextual computing for all kinds of mobility-related scenarios is huge, and therefore tourism and tourists can greatly benefit from a rigorous and inherently enabling approach to the exploitation of context information, which in turn will boost their tourism experience[Sig06].

In addition to the conceptual and theoretical motivations found during the review of the literature, there are also other motivating forces. The market of communication technologies providing access to information *en route* is one of the most promising of all existing markets, showing the greatest growth and evolution. The following list indicates some of the results and forecasts extracted from studies conducted by organizations specializing in the realm of communication technologies:

1. There are about 4.1 billion mobile subscribers Worldwide, while there are just 1.3 billion land-line subscribers (Fig.1.3). This study further indicates that in areas where there is no *Legacy Market* the trend is to go mobile. For example, 28% of the African population owns a mobile telephone, whereas only 2% of the population uses landline telephones[11];

2. The 25 greatest mobile technology operators account for more than 2.7 billion subscribers, i.e. around 64% of the total subscribers world-wide (first quarter 2009). Thus, just Chinese Mobile accounts for 479 million subscribers, whereas Vodafone (as second global operator) serves 247 million people in 19 different markets.

[11]Source: ITU, http://www.itu.int/newsroom/press$_r$*eleases/aboutitu.html*

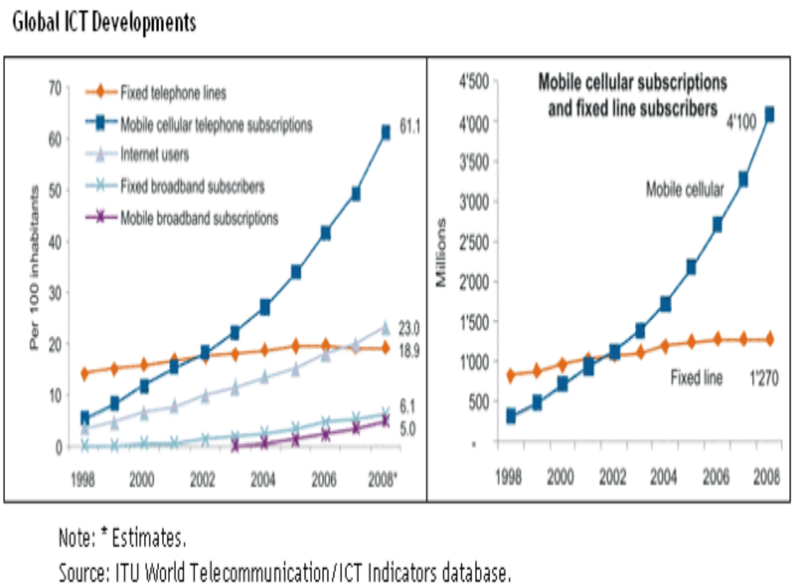

Figure 1.3: Number of mobile phones compared to land-line phones. Source: ITU

3. Globally, mobile data traffic will double every year through 2013, increasing 66 times between 2008 and 2013. Mobile data traffic will grow at a CAGR (Compound Annual Growth Rate) of 131 percent between 2008 and 2013, reaching over 2 exabytes per month by 2013. Mobile data traffic will grow from 1 petabyte per month to 1 exabyte per month in half the time it took fixed data traffic to do so. In the 7 years from 2005 to 2012, mobile data traffic will have increased a thousand-fold. The Internet grew from 1 petabyte per month to 1 exabyte per month in 14 years. Mobile data traffic (in comparison to voice services) will double each year up to 2013. Almost 64% of mobile data transfer by 2013 will be videos. Mobile video will grow at a CAGR of 150 percent between 2008 and 2013[12];

4. Intelligent mobile phones (Smartphones) represent 10% of the market. This figure is foreseen to increase up to 40%-50% in the next 5 years;

[12]Source: Cisco Systems

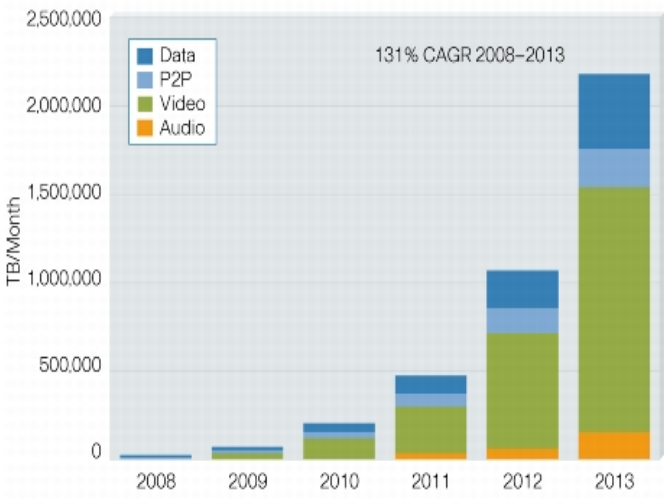

Figure 1.4: Mobile data traffic forecast. Source: Cisco Systems

5. 3G and high speed broadband mobile device access will generate around 80% of mobile data by 2013. Thus, one single Smartphone, such as the iPhone or a Blackberry will generate more traffic than 30 first generation mobile devices. A laptop with broadband access will generate over 450 times the traffic of an average mobile device;

6. The total amount of broadband users will increase up to 1025% to 2 billion by the end of 2014;

7. Global annual income in mobile telephone communication will be of around 1.03 billion US dollars in 2013;

The theoretical limitations in the existing approaches to human mobility together with the overwhelming figures and trends detected both in the tourism and telecommunication sectors, are the main motivation to rethink the study of context-awareness, despite its rather long tradition in computer science research.

1.2 Push context-aware systems in tourism

As mentioned earlier, tourism is a particularly well-suited domain application for contextual computing services, since tourists can greatly benefit from automatic assistance through mobile devices while they are at their destination.

Mobile tourist guides have come a long way since the earliest examples[AAH+97]. At first, the main objective behind research was to investigate how to implement working prototypes to identify (and help overcome) the potential problems of mobile tourist guides. Due to technology advance, these first problems seem to have been solved and there are some first commercial products available[GPW+08].

The second generation of mobile tourist guides specializes on features such as personalization, recommendation and location-based services together with new forms of user interaction. Most recently, due to the Web 2.0 phenomena, mobile tourist guides also include collaborative usage and social integration features that allow a particular individual to contact with other individuals with similar demographics located nearby, even if they do not know each other.

According to Grün and colleagues [GPW+08] most mobile tourist guides either provide location-based information, such as sorting the search results based on the location given by the device, or concentrate on delivering personalized information, such as filtering restaurants based on user preferences or demographics. What is still missing is the combined usage of various context factors, e.g. location and user preferences, device and network, environmental conditions and tourism services within the environment in order to provide more relevant information to visitors.

In addition, push information services are particularly appropriate for tourism, since it relieves tourists from having to actively search for the information they require in devices with small screen and keyboard. According to the literature review, only two push context-aware applications have been identified in the tourism domain, i.e. etPlanner[HFZ+06] and CAIPS[BFH+07].

Thus, from an epistemological point of view, push contextual computing information systems are still an open issue.

1.3 Digital Broadcasting as mobile communication technology

The tremendous increase on 4G, 3G, UMTS, WiMAX and GPRS/HSDPA communication technology-based download and streaming services, and the decrease in price of hard memory in mobile storage and player systems (e.g., iPod[13]), give rise to the following questions: does any other communication technology make sense? Would particular sectors (due to their features and characteristics) require another kind of technology that better suits their nature? Would in this case, due to its *push* nature, digital broadcasting-based information-intensive services make sense? Would digital broadcasting be an appropriate communication channel in order to distribute tourism information to support human mobility?

In this sense, two trends have been identified. One of them is related to migration of connection services to 4G technology in mass markets in order to reduce operational costs. These are on-demand services and require active user interaction to generate more data transfer on the network, i.e. the user is actively looking for information to download. These kinds of services focused mainly on mass markets and are the typical example of *pull* based or *lean-forward* services. One of the key factors for success of these services is the great initial investment in infrastructure to guarantee a high quality of service (QoS). Nowadays, 3G data service distribution systems are reaching saturation in markets where there is a high subscriber density. For example, Verizon network subscribers in some cities in the US and DoCoMo network subscribers in Japan have experienced some network collapses due to data traffic overload. In these cases, migration to 4G based systems is essential in order to maintain the QoS and to be able to support the enormous amount of current data traffic and the massive increase of traffic expected in the short and medium terms.

The second trend is digital broadcasting (both terrestrial and satellite) services, as well as the evolution of broadcasting as complementary technology for analogue broadcasting. Digital broadcasting systems aim at maintaining and, if possible, increasing their market share by providing high quality sound, grater ease of use for the final users (stressing on one of their key-issues for success, i.e. ease and convenience of use) as well as providing additional data services that generate a differential added value and a greater attraction for the user.

These kinds of services are well suited for markets complementary to mass-consumption connection-based markets. Digital broadcasting services aim to easily and conveniently fulfil the requirements of *push* niche markets in which preselected, up to date and time-distributed content is what the user really desires. Thus, the automotive, transportation, traffic information, tourism (information-intensive) and entertainment niche markets are more adequate for digital broadcasted consumption schema than interactive (*pull*)

[13]http://www.apple.com/es/ipodclassic/

services based on a real connection. Definitely, the cities and the diverse environments will be structured under different technological frameworks, such as the Future Internet, Cloud Computing, Ambient Technologies, digital broadcasting and so on.

1.4 Work goals, hypothesis, assumptions, limitations and delimitations

Based on what has been explained in the previous sections (see Section 1.1, Section 1.2 and Section 1.3), **this dissertation aims to**: Firstly, **construct a theoretical framework to model context** and, secondly, to **develop an application to study the extent to which digital broadcasted tourism information can be contextualized so as to provide context-based information which is relevant to support human mobility**.

In order to do this, the content must be first accessed and retrieved; then the information must be appropriately adapted for broadcast. Following, the information is broadcasted and received by the mobile device. **This dissertation focuses on the process of filtering incoming broadcast information to the mobile device**. Therefore, the incoming information needs a context model and a set of rules adapted to the *push* nature of information distribution as well as to the needs of niche markets that require preselected content, e.g. tourism. Thus, visitors may potentially be better served by digital broadcasting services rather than by traditional connection systems.

Thus, the general research questions that have guided the piece of work proposed in this dissertation are:

- *How can contextual computing information based applications be implemented with minimal infrastructure and deployment overload?*
- *Can ontologies be used to filter incoming push broadcasted information according to the values of context at a given location and time?*

Considering these two questions, this dissertation describes a **Web information-based, visitor-centred context framework for semantic-based contextual computing applications to support human mobility. Rather than areas populated with sensors, the Framework uses mobile embedded sensors, such as RFID or GPS and disseminates both context and tourism information through digital broadcasting** (see figure 1.5) taking into account the following premises:

- The Internet as primary source of information for both context and tourism information.
- Digital broadcasting as communication technology.
- A semantic-based rule filtering engine that provides visitors with relevant information according to their context. The values of the different constituents of context at a given moment are the filter for the rest of the context information.

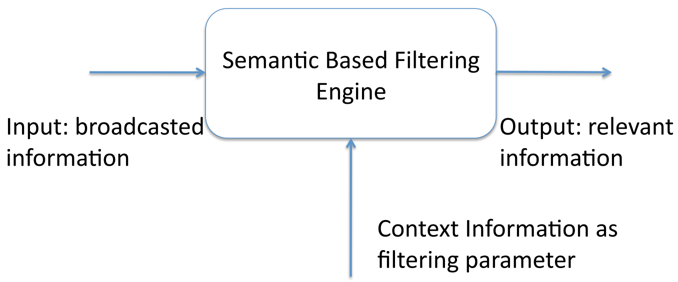

Figure 1.5: System draft

The following hypothesis (subdivided in four sub-hypotheses) is proposed in order to provide an answer to the general research questions:

- *H(1)*: A push contextual computing framework can be established by gathering context information from Internet information sources without external sensors.
- *H(2)*: Digital broadcasting can be used to disseminate context and tourism information to mobile devices equipped with a filtering engine designed to support human mobility.
- *H(3)*: A network of ontologies approach can be used to model visitor context and use it as a tourism information filtering engine based on description logics (DL) and rule-based reasoning.

- *H(4)*: This framework increases user satisfaction by using a semantic-based visitor-centred model for context, which is based upon recommendations provided by the UNWTO on visitor classification.

A complete semantic-based contextual computing model and architecture have been designed, modelled and implemented in order to validate the hypothesis. Thus, the general goals of the dissertation are the following:

- *GOAL 1*: To design and model a push contextual computing framework for human mobility that results in a new definition of context and that is centred on the mobile nature of people. (Section 3.1.2).

- *GOAL 2*: To gather contextual and tourism information from Web-distributed and heterogeneous sources of information and disseminate it via digital broadcasting. No external sensors shall be used, other than those embedded in mobile devices. Content gathered in the Web will then be adapted for broadcast. This will constitute the first level of interoperability proposed by the new framework, i.e. interoperability at the infrastructure level. (Chapter 3).

- *GOAL 3*: To design, model and implement an ontology-based model of context, using networks of ontologies in order to increase the models consistency, reasoning capabilities, modularity, interoperability, re-use and sharing. The ontology will particularly focus on modelling the visitor according to tourism-established vocabulary and will implement a rule-based reasoning engine to filter incoming information. This will constitute the second level of interoperability proposed by the new contextual computing framework, i.e. interoperability at the model level. (Chapter 4).

The following specific goals originate from the general goals:

1. To study the notion of context as a scientific research topic on its own within the framework of human mobility and not as an auxiliary tool for other research disciplines such as Human Computer Interaction[Dey00], Distributed Computing[Str03] or Intelligent Software Agents[CFJ04b], to name a few. The significance and nature of context in tourism are relevant enough to merit research on the application domain of context. This new approach ought to generate a new operative and integrative definition of the notion of context.

2. To place the human being, the visitor in this case, at the centre of the problem. Existing approaches have studied the context of an

application or a system. This dissertation proposes to study the context of the traveller.

3. To use the Internet as the main source of contextual information. Web originated information could eventually be enriched with mobile device incorporated sensor information, providing the mobile device is equipped with these kinds of systems.

4. To make Contextual Computing applications universal: by changing the perspective of the notion of context (i.e. centred around human mobility) and by avoiding networks of sensors or using only mobile device embedded sensors and Web-based information architecture.

5. To use alternative communication technologies such as digital broadcasting to deliver contextual computing services in tourism, which come along with the nature of this business sector (Section 1.3).

6. To follow the new philosophy and approach to the notion of context in tourism, provide a network of ontologies that could be put forward as one first step towards standardization of a Context Model.

7. To develop a simple yet effective architecture to provide Contextual Computing services to support human mobility with minimal programming overload.

8. To develop a simple yet flexible and scalable location independent infrastructure.

These specific goals focus on the need to study context from a completely new perspective and address one of the major constraints of current context-aware approaches: the use of sensors to gather contextual information, which makes context applications highly dependant on the area that has previously been populated with sensors.

1.5 Research Methodology

Experience shows that the design, development and evaluation of contextual computing based applications is complex[Ay07] [GWPZ04] [CFJ04b] [BBH$^+$09] due to the reasons shown in previous sections, their highly distributed nature and the degree to which infrastructure work importantly limits the use of the future application. Interestingly, the research methodology on its own is being thoroughly studied by the research community[HL09].

Similar research objectives can be approached in completely different ways depending on the research context: availability of infrastructures, accessibility and proximity of experts, synergies with ongoing research projects and so on.

Based on the context of this dissertation, a research strategy was designed to carry out the following activities:

1. Update knowledge by reviewing state-of-the art publications on the realm of context-awareness, Semantic Web and ontologies, and tourism mobile applications. This knowledge has been reinforced attending specialized conferences.

2. Establish a critical evaluation (scope, limitations, advantages and disadvantages) of current approaches in the realm of context-awareness in general and within the framework of human mobility in particular.

3. Design and develop the different parts of the prototype and the architecture augmenting the scope and functionality in interactive processes and developing each of them as separate independent modules.

4. Experimentation and evaluation of the prototype and its components at each stage.

5. Attend conferences and workshops to present partial results and validate them with the scientific community and check and analyse existing state-of-the art development.

6. Network with experts at conferences, meetings, via email and visiting other research centres[14].

7. Redesign with feedback from all the above described processes.

8. Development of the CONCERT prototype based upon the theoretical work carried out following the previous steps to support Human mobility by means of a semantic-based contextual computing system through digital broadcasting.

[14]The author was a visiting Ph.D. student at the Institute of Software Technology and Interactive Systems at the Vienna University of Technology for three months (September - December 2009). The author was a member of the eCommerce group lead by Prof. Werthner where similar projects within the realm of context-based tourism mobile applications are being carried out at the moment.

9. Technical and user evaluation of the prototype.

10. Dissemination of the results obtained throughout the research process in order to contrast with the peer community the research results of this work. (A list of publications is available in Section 7.3).

Figure1.6 shows schematically the methodology that has been followed to accomplish this work.

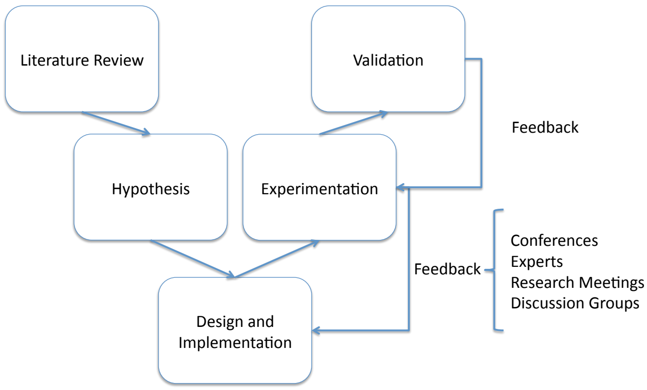

Figure 1.6: Research methodology

1.6 Dissertation outline

This dissertation is structured in seven chapters and a number of annexes as follows.

Chapter 1 (this current chapter) contains the general introduction to the dissertation. It presents the starting concepts, i.e. human mobility, technology and context and the way in which they are linked with each other. It also shows the relevance of contextual computing services in tourism and of digital broadcasting as an alternative for push-based information intensive service niche markets. The problem description and motivation, research hypothesis and main objectives are also shown to clearly limit the scope of this dissertation. Finally, this chapter shows an overview of the research methodology followed in the development of this work.

Chapter 2 presents a thorough review of relevant literature and theory. Evaluation criteria have been established in order to identify and describe the research work carried out up to now in context-aware computing. This has helped to clearly identify the gaps that need to be filled in context-aware research in order to try to make these applications available to the general public, in particular in such a distinct discipline like tourism. The related work tackles primarily context-awareness related work. It identifies two trends and analyses their main characteristics primarily differentiating them by their theoretical work defining the foundations of the notion of context and the use of semantics to manage context information. It also deals with the work done by the Artificial Intelligence (AI) community in the area of Knowledge Representation (KR) and how this work has lead to the use of ontologies in different domains included ubiquitous or pervasive computing. Since the CONCERT framework context model is based on a network of ontologies, Chapter 2 also contains a review of the most relevant existing context ontologies, some of which have been (at least partially) re-used in the ontology developed in this work, as shown in Chapter 4. Finally, Chapter 2 reviews some of the most representative examples of mobile tourism guides and shows to the extent to which the work presented in this dissertation extends the previous examples as well as its main contributions.

Chapter 3 presents the conceptual framework of this dissertation and its most important theoretic contributions. It starts by addressing the new approach to context. Following, it presents the two level interoperability concept. It then shows the implications the two level interoperability concept has in the traditional approach to the theory of context and suggests new definitions of some terms in order to on the one hand, set an operational definition of diffuse concepts and on the other hand, set the scope of contextual computing in tourism. Following, the context information requirements are reviewed and updated according to the implications of the new theory.

Chapter 4 presents one of the most important parts of the context model: the context network of ontologies named ContOlogy. The chapter begins by presenting a number of methodologies for building ontologies that are available in the literature. It then establishes the reasons for choosing one of these methodologies and details the process of ontology building according to the chosen methodology. The Chapter concludes with the graphical representation of the ontology and its thorough description as well as with the validation of its functionality according to its definition requirements.

Chapter 5 presents the architectural support for the CONCERT framework. Firstly, it presents the CONCERT framework context architecture and its main characteristics. It then reviews a general architecture of digital broadcasting and describes the various standards available and concludes by justifying the reason why DRM shall be used for the purposes of this dissertation. Special sections in this chapter pay particular attention to the structure of content for digital broadcasting. A combination of digital broadcasting and CONCERT architectures is then presented and following, the Chapter illustrates an application that has been developed following the CONCERT guidelines and describes its different components, i.e. the CIC RSS Content Maker and the CIC Viewer. The concluding part of this Chapter is particularly relevant, since it demonstrates the extent to which the research questions and the first two sub-hypotheses have been answered.

Chapter 6 shows the evaluation of the CONCERT framework. The Chapter is further subdivided in two sub-chapters. Firstly, the Technical Evaluation of the CONCERT framework is shown. After having determined the critical variables that are going to be measured, several tests have been carried out in order to see the relation of these variables with each other. The tests have been run in different machines and a qualitative analysis is shown. Then, a User Evaluation has been carried out to see the acceptance the model would have in the public. The empirical results obtained from the evaluations have been used to corroborate the second two sub-hypotheses.

Finally, **Chapter 7** shows the major conclusions of this of research. It also shows how the objectives have been reached and the degree in which they have been fulfilled. Next, some future research lines are proposed. The Chapter and thus, the dissertation conclude with some final remarks.

Chapter 2

Related Work

"The more extensive a man's knowledge of what has been done,
the greater will be his power of knowing what to do."
Benjamin Disraeli, (1804 - 1881), British Prime Minister.

Since research on context-awareness began now 20 years ago, intensive efforts have been dedicated to do research within this discipline and as a consequence, there exist an extensive theory and number of context-based systems and applications. The objective of this chapter is then to present a descriptive and analytic summary of the state of the art.

Firstly, the evaluation criteria followed to analyse theory, concepts and systems is presented in Section 2.1.

Research on the realm of context-awareness can be divided into two trends. Thus, Section 2.2 presents the work carried out by researches from the first trend during the 1990s. This work was basically focused on defining the theoretical and conceptual foundations of context-awareness. In fact, the theoretical work carried out was so intense that the most relevant definitions of 'context' were put forward throughout those years[Wan92][SAW94][RPM97][HNBR97][Sch99][DA00b][CK01][Pas01] [Sch06].

None of the context-aware systems initially put forward suggested the use of semantic technologies to manage contextual information. At that time (1990s), the potential and functionalities of ontologies[Gru93] were still not clearly specified and therefore, researchers did not even consider them an alternative for context information management. Thus, they used different methodologies[SLP04][BBH+09]. This way, Section 2.3 is devoted to present the basic notions of Semantic Web and ontologies.

Exhaustive analyses of context management methods[SLP04][BBH+09] indicate that ontologies comprehend adequate functionalities for context information management. Hence, several authors working on context-awareness converged upon a fact: semantic technologies could be employed to model context and manage context information. This would set the way for systems to more easily share, integrate, exchange and re-use context information and moreover, it would support not only to check the model's consistency, but also to infer implicit context knowledge [Ay07][CFJ04b][Str03][Bal07][BBH+09]. In this manner, Section 2.4 presents the work carried out by authors from the second trend of context-aware research, as of the year 2000.

Continuing with semantic-based context management, Section 2.5 presents and reviews the most significant context ontologies found in the literature and shows the terms that more frequently have been used in context modelling.

Context-aware based applications have been used in a number of domains including tourism, where mobile guides not only have a long tradition already[GPW+08][SGPR09], but represent also the opportunity for new generation tourism information-based products and services[BCGB07]. Section 2.6 will present some of the most relevant existing tourism applications.

Finally, the summary and conclusions presented in Section 2.7 will conclude the Chapter.

2.1 Evaluation Criteria

Some evaluation criteria have been established in order to systematically analyse, assess and compare the different theoretical approaches and applications in context-awareness as well as to better determine the contributions of the research work that is presented in this dissertation.

The selected criteria have been set up by observation of the features of the theories and applications that have been reviewed. The focus has been put on two aspects: Firstly, on the background expertise and research interests of authors (epistemologically essential to analyse the process of creation of new knowledge[Kli05]). Secondly, to the way the notion of context itself has been attempted. Experience has shown that the resulting context theory is highly influenced by these factors and further, by the particular objectives of a research project.

The rest of the elements that will qualitatively be assessed in this evaluation have been set comparing the features of the various applications examined during review phase. The most important characteristics are shown in the following list:

- *Definition and approach*: It refers to whether authors provide a (new) definition of the notion of context or rather they acknowledge another one given by someone else. In case there is a new definition proposal, the researchers' background expertise is also specified to better understand his theory.

- *Re-usability*: It refers to whether the application being reviewed is potentially re-usable outside the scenarios, areas, preconditions and assumptions of authors.

- *Sensors*: It refers to whether applications use (physical) sensors to gather context information or it is gathered otherwise, e.g. Web information sources through Web services.

- *User privacy*: It refers to whether or not applications have been enabled to protect user personal information.

- *Human intervention*: It refers to the amount of human intervention (both within developing and implementation phases as well as prior to application use by the final user) required to make applications work.

- *Environment*: It refers to the kind of space in terms of indoor or outdoor where applications may be used.

- *Context Model*: It refers to the methodologies and techniques used by authors to manage context information.

- *Technology*: It refers to the technology used for mobile device communication within the context model.

- *Nature*: It refers to whether the service provided is *pull* (active interaction with devices) or *push* (lean back kind of interaction).

2.2 Context-awareness (1^{st} Generation)

The history of context-aware systems started with the *Active Badge Location System*[Wan92][Bal07]. The notion of context in this work is exclusively restricted to the location of people within an office environment. However, the location of an individual is only one of a large number of variables that may be used to define context. Therefore, this conception of context turns out to be too broad and rather than a context-aware application, the *Active Badge Location System* would be a location-based application.

Accordingly, the applications that were built on top of this theory were conceptually very simple and provided just location-based related services in an indoor sentient environment. Efficient location and coordination of staff in many large organizations happens to be a recurring, difficult and challenging problem to solve. In order to elucidate this problem, a tag in the form of an Active Badge (see Fig.2.1) that emitted a unique code was built. The signal produced by the Active Badge was then picked up by a network of (infra red, IR) sensors placed around the host building and displayed in a useful visual form in electronic devices. People on the building were given Active Badges to carry during their stay indoors, so that the system could locate them. This way, telephone calls could be forwarded to the nearest device found close to the latest location identified for the user.

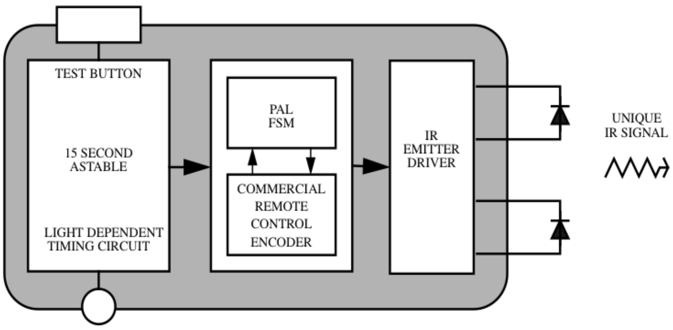

Figure 2.1: The ORL Active Badge

As well as the rest of the applications referred to throughout this Chapter, the *Active Badge* was conceived to be used *ad-hoc* under very concrete circumstances. In fact, this application could work under certain preconditions and assumptions: a given building had to be populated with sensors, certain people had to be provided with an Active Badge and their location could only be detected with a particular device. This poses serious barriers for scalability and actually goes somehow against the original conception of Ubiquitous Computing as formulated by Mark Weiser, i.e. anytime, anywhere and seamlessly[Wei99].

In addition to that, in all forms of Ubiquitous Computing user privacy is fast becoming a major concern [Wan92][Che03][Dey00], since systems and applications use very sensitive personal information in order to be able to deliver an accurate service. However, not much has been done in this regard so far. The *Active Badge Location System* is fully technology driven and it does not take into consideration humanistic aspects of the system. As already pointed out in the literature, privacy of users is not considered and fears of potential future misuse of it are revealed.

The term *context-aware* was first introduced by Schilit and Theimer[SAW94] to specify software that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time". They state throughout their work that humans live in a mobile and continuously changing environment in which interaction occurs with a number of different devices, sometimes at the same time. They take Weiser's[Wei99] Ubiquitous Computing vision idea into humans' context in terms that interaction always occurs within various circumstances (contexts) and locations that (in the examples provided) cover indoor environments.

People interact with a collection of devices that are communicating and cooperating on behalf of the user in what they call a *mobile distributed computing system*. With people moving within a constantly changing environment, the challenge is to (automatically) detect and exploit the changes that occur in the environment with a new class of applications that are aware of the context in which they are run, i.e. a context-aware application.

These authors show a wider notion of context as they assume that context is location as well as other important aspects such as *where you are, who you are with and what resources are nearby.* This statement provides some insight about their understanding of context, for they have no formal (explicit) definition of this concept.

Therefore, Schilit and Theimer's implicit definition of the notion of context could be the following[SAW94] :

> *"Context is who you are, where you are, who you are with and what resources are nearby as well as changes to those resources over time"*

Such a definition of context contributed to have a better understanding of the circumstances of the environment. The notion of context included in this case entities such as lightning, noise level, network connectivity, communication costs, bandwidth and even social situations. Being this a more comprehensive conception of context, the ambiguity comes from the fact that there are a number of terms that need to be further specified. For example, when they refer to "who you are", is it the name of an individual what they refer to? Is it rather the name and profile? The meaning of "nearby" is neither accurate enough and the extension (in terms of surface) covered by this notion ought to be clarified as well, for it is highly dependant on context, making the definition rather recursive.

All in all, this definition of the notion of context provides a considerable step forward in relation to the first one[Wan92], in spite of some more concreteness being needed. Context is not only the location of an entity any more. On the contrary, it also includes other important variables as well, that permit to more consistently understand the overall situation of that entity in question. As a consequence of considering more variables within the context, their context model is somewhat more complete than Want's and thus supports a wider functionality and range of services.

Schilit and Theimer[SAW94] developed The ParcTab application (Fig. 2.2) based on their theoretical conception of context that run in a small hand-held device. The ParcTab application used an infrared-based cellular network for communication (as well as the *Active Badge Location System*[Wan92]) and took into account more elements other than location, as derivation of their notion of context. The device acted as a graphical terminal and most applications run in a remote server, to enhance battery life and relief the mobile device of computation load.

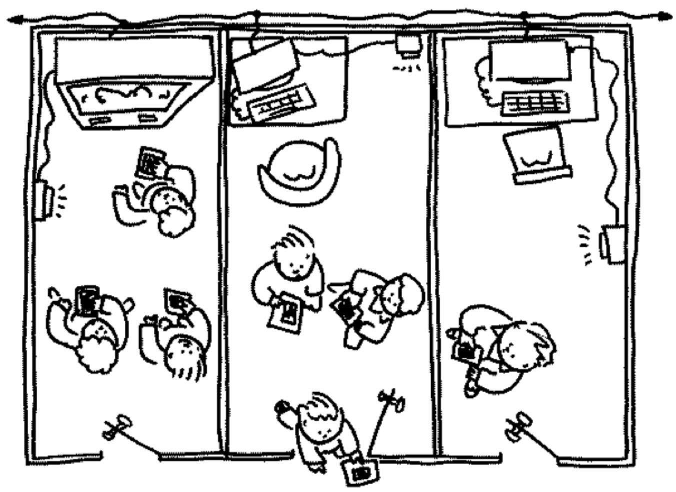

Figure 2.2: The ParcTab Context-aware Computing System

They classified four context aware applications:

- *Proximate Selection*: this is a user interface technique used to emphasise the location of nearby selected objects.
- *Automatic Contextual Reconfiguration*: Adding, removing new components or altering the connections between components, e.g. servers and their communication channels to clients.
- Contextual information and commands: people's actions can sometimes be predicted. There are certain things that we do when we are at certain locations, such as the kitchen, library and so on. This kind of application aims at exploiting this fact.
- *Context Triggered Actions*: a kind of IF-THEN rule engine that governs the adaptation of the application to changes occurring in the environment.

One of the most referenced definitions of context has been given by Dey and Abowd[Bal07][BBH+09]. Their definition of the notion of context is as follows[Dey00][DA00b] :

"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves"

This definition turns out to be relatively vague and its recurrent formulation makes it somewhat ambiguous. One of its weak points comes from the fact that the definition of context is put forward depending on the concept of situation and following, the situation itself depending on the context. For this reason, this definition is not optimal from a scientific point of view[Str03]. However, since it has been the mostly referenced definition of context, the ambiguity in its original formulation has originated a number of different interpretations according to the objectives pursued by the authors for their particular applications.

Surprisingly, in spite of being a widely accepted definition it does not focus on context itself. It rather focuses on the interaction of an individual with a device. That is precisely the backbone and key concept of their theory: interaction, not context.

In fact, they approach the notion of context from a Human-Computer Interaction background. They claim that in human-human interaction a lot of information is conveyed without explicit communication through context, whereas in human-computer interaction there is very little context (if any at all) shared between the human and the computer, which makes interaction with machines more unpleasant and harsher[DA00b]. Environmental or context information cover information that is part of an application's operating environment. Therefore, this environmental information could be sensed by the application itself[DA00a], thus turning this information into one more input parameter to the system[DSAF99].

Despite this may seem to (conceptually) ease the development of context-aware applications, the lack of uniform support for building these kind of applications has turn out to be a major problem[DAS99]. In an attempt to shed light on context-aware application development, they found an implementable way to handle this new input parameter, i.e. context information, through widget abstractions[DA00a]. In fact, context widgets (Fig.2.3) show some benefits, similar to graphical user interface (GUI) widgets: separation of concerns, re-use, easy access to context data through polling and notification mechanism and a common interface. Context widgets encapsulate a single piece of context and abstract away the details of how context is sensed.

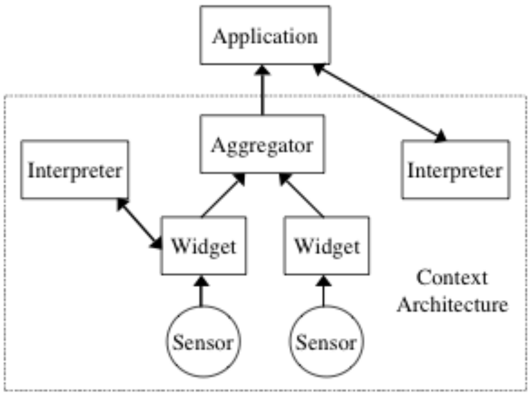

Figure 2.3: Sample configuration of context components. [DA00a]

Dey also provided a definition for *context-aware* as follows[Dey01]:

"A system is context-aware if it uses context to provide relevant information and/or services to the user where relevancy depends on the user's task."

Considering that Dey and colleagues argue that there is a lack of infrastructure support to make context-aware applications more present due to interoperability problems[DAS99], it may be surprising that they did not use ontologies (that provide particularly good information exchange and re-use features[Gru93]) for their context model (Context Toolkit). It is important to remark at this point that Dey and his colleagues were more interested in context data to support interaction and make applications react to certain external stimuli, rather than to process context data to deliver a particular service. Thus, no ontology reasoning (or other ontology functionality) capability is needed to reach their objectives.

Another important weakness of the Context Toolkit[DA00a] is its lack of scalability. Dey's Context Toolkit widget-abstraction only works with its programming code and there is a lot of work to do in order to make the sensors understand with their server, i.e. it is extensible provided the sensors are plugged into a system of its own. Else, the application does not work. Although they[DSAF99] claim to have built a re-usable solution, evidence shows that it is not the case. Their context widget and "building blocks" can only be used with their proprietary application and its (straight forward) utilization in other applications and even in theirs has not been shown to be obvious. The context widget theory is interesting from a conceptual point of view but unfortunately they are not to be programmed in a standard way and therefore the problem of interoperability still remains.

Dey and colleagues (as well as Want and others) have also shown some concern with user privacy issues. In fact, Dey's Context Toolkit[Dey00] introduces the concept of context ownership. Users are assigned to sensed context data as their respective owners. They are allowed to control the other users' access to their own data.

Another important shortcoming about Dey's and colleagues' application (also found in most of the examples available in the literature) is the amount of human intervention needed, especially when developing the application. Assembling the sensors, sensor communication channels, information reception, calibration and so forth, together with all the programming work and the lack of common understanding of the notion of context and different context models, makes the building of context-aware applications very complex. In addition to that, applications' dependability on context gathering infrastructure does not allow them to have a high degree of stability, since the failure in one single sensor may damage the whole functioning of an application and moreover, the component that failed has to be manually restarted[DSAF99].

G. Chen and David Kotz undertook a survey on Context-Aware Mobile Computing Research[CK01]. They reviewed a number of definitions that were available at that time in the literature and concluded that context as a general word has a loose definition, due in a great extent to its wide and misleading use. On the other hand, they understand that context in mobile computing environments has two different aspects:

- One aspect of context includes the characteristics of the surrounding environment that determine the behaviour of mobile applications. This refers to the so called active context.
- The other aspect of context is relevant to the application but not critical. It is not necessary for applications to adapt to the second kind of context except to display them (the general characteristics) to interested users. This refers to the so called passive context.

They argue that none of the formal definitions of which they had notice, distinguishes between these two kinds of context and thus, they proposed a definition of their own as follows:

"Context is the set of environmental states and settings that either determine an application's behaviour or in which an application event occurs and is interesting to the user."

Of course this definition is very interesting from the kind of context information point of view and the type of reactions it triggers on an application. However, it does not specify which set of environmental states determines particular behaviours on systems or applications nor it defines the kind of applications themselves. This definition is interesting from a possible application type in a context-aware system point of view. However, it is wide and vague in terms of definition specification and therefore leaves a lot of room for interpretation and consequently does not help to find a standard context framework. How could we take a step forward and develop framework efficiently?

Apart from the previous (considered to be the most relevant) authors, other researchers have also proposed their own definitions of context. In most of the cases, these definitions turn to be very general and thus, their use in a real computing model is very complex, for many (ambiguous and non-explicitly specified) variables could fit into those conceptions of context.

Schmidt and colleagues[Sch99][Sch02] have had a similar approach to context as Dey[DA00b]. They claim that context is a key discipline in Natural Language Processing but more generally and in particular within Human Computer Interaction. In fact, Schmidt and colleagues[Sch02] claim that general context information other than merely location can support a more implicit kind of interaction with mobile computing devices that may enhance the vision of invisible or disappearing computing, following Weiser's vision[Wei99].

In his work, Schmidt introduced a structured conception of the notion of context (very similar to the one suggested in this dissertation later in Chapter 3) and proposed the following model:

- A context describes a situation and the environment, a device or a user is in.
- A context is identified by a unique name.
- For each context a set of features is relevant.

A further classification of the context hierarchy is shown in figure2.4.

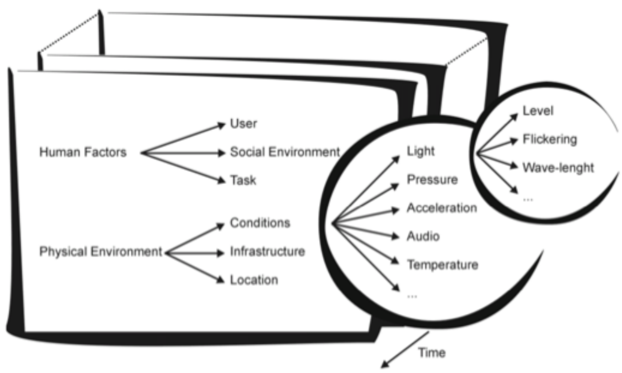

Figure 2.4: Context feature space

They distinguish between explicit and implicit context in the following way: explicit context is that coming directly from the human, i.e. the human explicitly specifies it in the mobile device. On the other hand, implicit context is the one that has been acquired by monitoring the user and computer-based activity[Sch99]. They propose to use sensors to gather contextual information, which happen to be the base of their context model. They model context by means of XML and on programming primitives.

Reconsidering the dimensions that complete the notion of context, Schmidt and his colleagues proposed yet another definition of the notion of context-awareness as follows[Sch02]:

"The knowledge about the user's and IT device's state, including surroundings, situation, and, to a lesser extent, location".

This definition turns to be very vague as it includes terms that need to be further interpreted. For example, with regard to "knowledge": who has the knowledge? What is that knowledge going to be used for? How? Such a definition is hardly implementable in a computing model in a general way.

In an attempt to elucidate who owns the knowledge on the context-aware system, Brown defined context as follows[Bro98]:

"Context is configured by the elements that are within the individual's environment and of which the computer has some knowledge".

This is a very wide definition as well and it is very difficult to unambiguously determine what the context of the user entails, and thus what the context model would be like. Furthermore, it is not clear whether the individual's characteristics and profile are also taken into account in this definition, which in turn are an essential part in the human-centric context conceptualization pursued in the framework put forward within this dissertation.

Very much based on Brown's Work, Pascoe[Pas01] defined context as:

"The subset of physical and conceptual states of interest to a particular entity".

Pascoe[Pas98] as well as other previous authors, approached the notion of context from an interaction perspective, in terms of interface behaviour of the wearable device. He proposed an extension to the interface that enabled computers to become aware of their context of use and intelligently adapt their activities and interface to suit the users needs at a given moment of time.

Ryan refers to context as[RPM97]:

"the user location, the environment or domain, his/her identity and date and time of the day".

The major drawback of Ryan's definition originates from the fact that this definition is too rigid and a somewhat vague. It does not allow an enlargement of the conception of context to new terms that could be relevant for the context. Some of the elements ought to be defined more in detail, such as what his understanding of 'identity' is, as well as how contextual information is going to be handled. He models the user's context with XML-based markup languages, in fact he created the ConteXtML language, which introduces a structure of concepts hierarchically organized.

Interestingly, Ryan explicitly considered the entity of time in his definition. Time has become one of the most relevant variables in context definition in last generation applications.

Finally, Hull refers to context as[HNBR97]:

"Aspects of the current situation".

This definition is also very wide and very difficult to abstract into a computing model. This definition seems to be close to the ones provided by ordinary dictionaries and does not allow to elucidate whether the current situation involves, for example, such important context factors as information about individuals, or their demographics and so on.

Last but not least, Pascoe also worked on context-awareness in an attempt to ease human machine interaction by extending graphical user interfaces, GUIs[Pas97]. He proposed an architectural support for context-aware applications that was based on the popular Post-It notes that can be stuck on desks, furniture, walls and so on with various purposes in order to provide information in a convenient way, or in an attention-grabbing location. The Stick-e-Notes he proposed, allow users to provide information electronically attached to a known location so that it can be of use for other users. His work is very much focused on location and there is no evidence of him considering other relevant context parameters. An important draw back of his application is that is it encoded on the C++ programming language, which does not make the application platform independent.

As mentioned at the beginning of this Chapter, despite all of the work shown, this first generation of researchers did not reach a consensus on delimiting the scope of context-awareness[HL09] and neither did they agree upon a unique methodology or model to manage context information.

2.3 Semantic Web and ontologies

At the end of the 90s and early years of the new century, the work carried out by the AI community since the early 70s on KR[New80] showed evidence that formal ontologies could be used as a way to specify content-specific agreements for sharing and re-using knowledge among software entities[Gru94]. Nowadays ontologies are considered (within computer science) a commodity that can be used for the development of large number of applications in different fields such as knowledge management, eCommerce, intelligent integration of information and information retrieval [CFLGP07].

In more recent years ontologies have also been extensively used in pervasive computing environments as a tool for developing context-aware systems [Str03][CFJ04b][GPZ94][Ay07]. Moreover, there have even been authors that claim that ontologies are key to the realisation of Context-Awareness[CFJ05].

Originally, the word Ontology (mind upper case 'O') [GG95] comes from philosophy (from the Greek, genitive $o\nu\tau o\varsigma$: *of being* (neuter participle of $\varepsilon\iota\nu\alpha\iota$: *to be* and $\gamma\iota\alpha$: *science, study, theory*). From a philosophical point of view, Ontology is the branch of philosophy that deals with the nature and organization of reality and things. It was the Greek philosopher Aristotle who first referenced this word when he defined a "science" that is "on top of" other sciences in his book Metaphysics IV, 1: a science that studies the being *qua* (i.e. as) being, i.e. Ontology[Ari C]:

"There is a science that studies the being as being and its properties as such (being) which belong to it in virtue of its nature. Now, this science is not the same as any of the so called special sciences, since none of these other treat (universally) the being as being itself but reducing the being to one part of it, they ("only") investigate the essential properties of this part. Since we are seeking the first principles and the highest causes, there must (clearly) be something to which these belong in virtue of its own nature. If then, those who sought the elements of existing things were seeking these same principles, it is necessary that the elements must be elements of being not only by accident but just because it is being. Therefore, it is of being as being that we also must grasp the first causes".

More recently, within computer science, ontologies (mind lower case 'o') [GG95] aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be re-used across applications [CJB99][CFLGP01]. Ontologies first started to be used back in 1991 within the context of the DARPA (Defence Advanced Research Projects Agency) Knowledge Sharing Effort[CFLGP07].

The origin of this work was in the efforts the Artificial Intelligence Community was doing at the time to find new ways to share knowledge. In fact, the objective of that project was to explore new ways to construct knowledge-based systems by assembling re-usable components, saving this way time and money. System developers would then only need to worry about creating the specialized knowledge and reasoners for the specific task of their system. Thus, declarative knowledge, problem-solving techniques and reasoning services could all be shared among systems.

Since the mid seventies, researchers have recognized that capturing knowledge is key to building large and powerful AI systems and applications[NFF+91]. Of course, one of the most complex problems researchers had to face was the need to represent the captured knowledge so that they could make some sort of meaningful understanding about it and set the rules under which it could and ought to be shared and re-used across applications. The problem of knowledge representation and sharing has been widely studied by authors like Allen Newell[New80], Nicola Guarino[Gua95], Gruber[Gru93][Gru94], Musen[Mus92] and many others.

This same grounding rational is precisely what underlies the context framework introduced in this dissertation: to provide the way in which (at least) part of the context (the individual's context) can be shared with other kinds of context (domains of reality) and thus, provide effective context based information services in tourism. This concept will later be introduced as *interoperability at the model level* in Chapter 3.

As well as is the case with the notion of context, there have been a number of definitions of the notion of ontology[GPFLC03] (in the domain of Computer Science) put forward in the literature. Similarly, their meaning tend to be a bit vague, since they have been used in a number of different ways and by authors with different background expertise and research objectives.

One of the first definitions was actually given by Neches and colleagues[NFF+91] within the DARPA project. They defined ontology as follows:

"An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary".

It is noteworthy in this definition that it does not only consider the terms explicitly defined in it to represent a particular domain of interest, but also the ones that can further be entailed in ways that are not specified within the definition itself. This way, the notion of ontology is given a wider scope and by extension, so is the domain it aims to represent. This makes it difficult to really know both what an ontology is as well as to delimit the domain the ontology covers. In addition to that, the main shortcoming of the given definition is the lack of references to the attributes, which are one key component of ontologies.

The most referenced definition of the notion of ontology within computer science is that given by Thomas R. Gruber. Gruber extensively worked within the AI community and therefore, his definition and work on ontology (engineering) is necessarily very much influenced by this research community.

Gruber argued[Gru94] that a body of formally represented knowledge is based on a conceptualization, whereby a conceptualization is an abstract and simplified part of the world that needs to be represented. Therefore, he defines ontology as follows[Gru93][Gru94]:

"An ontology is an explicit specification of a conceptualization".

This definition was proposed as the official definition of ontology within the AI community. Gruber argues that for an AI system, that what exists is that which can be represented, and representation of bodies of data were the key to show (and share) knowledge.

Although this is a widely accepted definition, it is quite specific for the AI knowledge representation problem and challenges, in spite of Gruber's arguing that the real objective of ontologies (and his overall work) is to set the way to share and re-use knowledge. In addition to that, this definition poses a debate in two of its nine terms, i.e. "specification" and "conceptualization". They have been given different interpretations that need to necessarily be clarified for one unique and unequivocal understanding of the notion of ontology to exist. However, this clarification would open a debate, which is totally outside the scope of this dissertation, since ontologies will be used as tools to model and manage context information and not as research matter itself.

One of the aims that underlies knowledge sharing and re-use is to enable people, organizations and software systems to more effectively communicate between and among them. However, their diverse background contexts and assumptions may impact the shared understanding subject matter. In particular, within the framework of computer science, this lack of understanding has lead to difficulties in identifying requirements and thus, in the definition of a specification of systems. The way to address this problem is by working on common and shared theoretical frameworks.

Along with this, Uschold and Grüninger studied ontologies under this approach and provided the following definition[Usc96a]:

"Ontology is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve problems, e.g. semantic interoperability, structuring and representing relevant concepts in a large knowledge base, etc.".

A conclusion that can be extracted from this definition is that one of the common objectives that all authors pursued with the use of ontologies was to find the way to exchange and re-use knowledge concerning a particular domain of interest. However, they all refer to the same concept using different words. Neither Gruber[Gru93] nor Uschold[Usc96a] explicitly refer to the extensions of the vocabulary or reasoning capabilities of ontologies. They are more focused on the KR problem, given their AI tradition.

Another interesting main idea that may be drawn from this conclusion is that this definition implicitly allows to guess that Uschold's and Grüninger's main objective attempts to establish shared understandings of domains of interest to enhance communication between people with different viewpoints and needs. Interoperability between systems (an increasingly relevant problem within the Travel and Tourism industries nowadays, by the way) and system engineering benefits in terms of reliability of software entities and specification of requirements.

Interestingly, Grüninger provided later on another definition of ontology in a work that he carried out together with Mark S. Fox[GF95]. In this latter work they developed a system to support reasoning in industrial environments. In fact, the reasoning capability ontologies provide to semantic-based computing systems is what guides their interests and work. In this case, their definition of ontology is as follows:

"An ontology is a formal description of entities and their properties, relationships, constraints, behaviours".

It would be interesting to find out, whether the noticeable evolution of the term ontology assumes that the first objective of achieving a shared understanding by the use of ontologies has been accomplished[Usc96a] and therefore, this other definition better suits the real meaning of the notion of ontology. Moreover, there are several other aspects of this definition that have been let out for personal consideration, such as the meaning of formal and behaviours. One could possibly guess that "formal" refers to the ontology representation language, whereas on the other hand, there is no way to know what "behaviours" may mean. However, it is interesting about this approach that the authors consider both the properties and relations of entities within the definition of the term, as main components of ontologies to pursue reasoning, contrary to previous authors.

Another very important author in the field of ontologies is Nicola Guarino. Guarino and his colleague Garieta[GG95] focused their interest in the philosophical nature and origin of the term. They did not phrase one formal definition of ontology, however they made the next understanding of the concept[GG95]:

"A set of logical axioms designed to account for the intended meaning of a vocabulary".

This definition does not leave clear what an ontology is. One needs to deeply read Guarino's work to really understand what his understanding of "intended" and "vocabulary" is.

What is relevant about Guarino's work in this field is the distinction he set between the notions of Ontology (upper case 'O') and ontology (lower case 'o'), yet maintaining a link between them. The word ontology shall be used with lower case 'o' to indicate the semantic tools used in computer science and with upper case 'O' (i.e., Ontology) to designate the discipline of philosophy. The way the term ontology is used throughout this dissertation in fact, will follow this distinction.

Other authors, no so frequently referenced throughout the literature, have also provided additional definitions of ontology. For instance, some years later, Borst slightly modified Gruber's definition and he defined ontology as follows[BBWA97]:

"Ontologies are defined as a formal specification of a shared conceptualization".

It is remarkable about Borst's definition that while he (almost literally) follows Gruber's definition of ontology[Gru93], he adds the term "formal" to it on the one hand, to provide a certain degree of formality (presumably again referring to a particular language of knowledge representation), and he follows Uschold's and Grüninger's definition by adding the notion of shared understanding on the other. However, this definition still shows the same ambiguities and weaknesses as Gruber's[Gru93], since the words "specification" and "conceptualization" still have open meanings, for they still have not been clarified.

In an attempt to disambiguate Gruber's interpretation, these last two definitions were later merged into another one and further explained by Studer[SBF98]:

"An ontology is a formal explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable. Shared reflects upon the notion that an ontology captures consensual knowledge, that is, it is not privative of some individual, but accepted by a group".

This definition turns out to be too long and rather recurrent, since it uses the terms to be defined within the definition itself, which makes the concept not very clear, for the defined term ought not to be part of the definition itself.

Finally, Voß[Vos04] defines ontology as:

"A formally defined system of concepts and relations between these concepts".

Summarizing, most of the definitions put forward are rather complex to understand at first sight. However, there has been a gradual evolution towards a more clear phrasing: while early definitions show a rather abstract AI formulation, the last ones focus more on their operational nature, making the definition more concrete and easier to elucidate, closer to the notion of knowledge sharing rather than to knowledge representation.

The evolution on the phrasing of the definition of the notion of ontology has also possibly happened for the free interpretation allowed in the meanings of the definitions. In addition to that, other important factors in that matter have been the intention of use of ontologies in the different application domains where they have been implemented, authors' knowledge background, objectives and interests[Kli05].

There has been (and there is) a lot of discussion promoted by Guarino[Gua95], McCarthy and Hayes[MH69] and other authors about the philosophical origin of the term ontology. They have tried not only to analyse the term, but also to analyse the various existing definitions and finally, put one of their own forward. And finally, there are other authors that see ontologies as simple tools that allow to represent a particular domain of interest that enables inference and reasoning.

Tables 2.1 and 2.2 summarize the review of ontologies and introduces the definition of the notion of ontology that is put forward later in Section 3.1.3.

Author	Definition	Background	Objective
Aristotle	There is a science that studies the being as being and its properties as such (being) which belong to it in virtue of its nature. Now, this science is not the same as any of the so called special sciences, since none of these other treat (universally) the being as being itself but reducing the being to one part of it, they ("only") investigate the essential properties of this part. [...] there must (clearly) be something to which these belong in virtue of its own nature [...] Therefore, it is of being as being that we also must grasp the first causes.	Philosophy	The study of the organisation and reality of nature
Neches et al.	An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.	Computer Science	To find a common way to define domains of interest.
Gruber	An ontology is an explicit specification of a conceptualization	AI	Knowledge Representation and Sharing
Uschold and Grüninger	The term used to refer to the shared understanding of domain of interest which may be used as a unifying framework to solve problems	Computer Science and Knowledge Engineering	Knowledge sharing
Gruber	An ontology is an explicit specification of a conceptualization	AI	Knowledge Representation and Sharing

Table 2.1: Table of ontology authors (I)

Author	Definition	Background	Objective
Grüninger and Fox	An ontology is a formal description of entities and their properties, relationships, constraints, behaviours.	Computer Science	Knowledge representation and Reasoning
Guriano and Garieta	A set of logical axioms designed to account for the intended meaning of a vocabulary	Computer Science	Clarity of terms.
Borst	Ontologies are defined as a formal specification of a shared conceptualization.	Computer Science	Formality and knowledge sharing.
Studer	Formal explicit specification of a shared conceptualization. Conceptualization refers to an abstract model [...] Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers [...]should be machine readable. Shared reflects upon the notion that an ontology captures consensual knowledge [...]	Computer Science	Knowledge Representation and Sharing.
Voß	A formally defined system of concepts and relations between these concepts.	Computer Science	Simplicity and Knowledge Sharing.
Lamsfus et al.	**Ontology is a set of formally defined concepts, attributes of these concepts and the allowed relationships among them that define a particular domain of interest. The formality and expressivity of an ontology is given by its representation language**	**Tourism Technology and Computer Science**	**Context Modelling, reasoning and Content interoperability**

Table 2.2: Table of ontology authors (II)

2.4 Context-awareness (2^{nd} Generation)

As of the year 2000, there have been numerous attempts within the second trend in context-aware research aiming at establishing a standard context management model, fundamentally by use of semantic technologies. In fact, this is one of the greatest differences between the authors of this trend and the ones from the first trend: the use of semantic technologies to model context and manage context information.

There are a number of examples of such context ontologies[CFJ04b] [GWPZ04] [WZGP04] [SRA06] [PdBW+06] (which will be reviewed in more detail in the following Section) in the literature. However, none of these ontologies fully copes with the requirements of the tourism domain and mobility, since assisting humans on the move in a large scale was not within the specific objectives of the projects in which they were developed. Moreover, none of these ontologies entailed neither tourism specific established vocabulary to model the entities that configure them[UNW08a] nor fulfilled the requirements of human mobility framework.

Another significant difference between the two trends is that the primary interest within the second is to set a standard context model and management methodology in order to ease the development of context-aware systems. Authors in this trend are not as interested in providing new definitions for context (although there were several attempts) nor in studying its theoretical and conceptual foundations as authors from the 1^{st} Generation. As a matter of fact, they all end up utilizing Dey's definition of context[DA00b] and they build their semantic model on top of it.

For instance, the notion of context has also been studied from the point of view of interoperability of distributed systems. In actual fact, any context-aware system is necessarily a distributed system, for it needs to process data originated in different (i.e. distributed) parts of a particular environment outside the system itself to act accordingly. This is the case of Thomas Strang and colleagues[Str03] [SLPF03] [SLP04] that studied interoperability of distributed systems, particularly focusing on interoperability at a context level[Str03].

They thoroughly analysed Schmidt's[SL01], Chen and Kotz's[CK01] and Dey's[DA00b] definitions of context and concluded that both Schmidt's and Dey's definitions on their own are not self-contained, but complementary. On the one hand, Schmidt proposes a definition that describes context depending on certain characteristics that are not specified. On the other hand, Dey provides these characteristics the level of "entity" in his definition, whereby entity is a person, place or object.

Since both authors come from the human computer interaction community, it is reasonable to think that they use the term "entity" with the same semantics and meaning, so that the combination of their definitions could be suggested. However, this extent has not been found in the literature, which does not provide evidence to Strang's assumption.

Thus, his definition of context is as follows[Str03]:

"Context is a set of all the context information relevant for a task".

Since the formulation of a definition cannot contain the defined term, he then defines context-information as follows:

"Context information is any information that can be used to characterize the condition of an entity, in terms of characterizing an Aspect".

As well as Ryan[RPM97] defined an XML-based context markup language (ConteXtML), see Section 2.2, in order to model context, Strang proposed to use an ontology-based language[SLPF03]. Modelling context is always a challenge, since all concepts and relationships need to very clearly be stated in a traceable manner. All parties in a service interaction at the context level need to have a common and shared understanding of the information they interchange, which they do through ontologies. However, they compared existing ontology representation languages and determined that if a language is well suited for representing knowledge, then it is not appropriate to support knowledge querying and vice-versa.

With this background, they defined the *Context Ontology Language (CoOL* as a collection of several fragments grouped in two subsets, i.e. *CoOL core* and *CoOL Integration*.

At this point it is mandatory to mention Harry Chen, one of the most popular authors in the realm of context-awareness. He described a framework for an agent based pervasive computing environment[CF03]. His approach relies on intelligent software agents to automatically perform various tasks[CFJ04b], allowing this way (human) users to focus on other (more relevant) issues.

Chen and colleagues did not deliver a notion of context of their own. However, possibly in an attempt to establish yet a new definition of context, in early papers they show the concepts they took into consideration[CF03] when dealing with context. Since they did not explicitly and formally phrase a definition of context, a semiformal one could be as follows:

"Context is an understanding of location and its environmental attributes, people, devices, and objects in the location and the software agents it contains".

This definition of context reminds in a way to the one proposed by Schilit[SAW94] (see page 23) with an additional reference to software agents, which are however subsequent to Schilit's time. In this manner, Chen's definition of context presents similar problems to Schilit's. For example, depending on the objectives one seeks, one may consider different environmental attributes, devices or objects within one same location. Hence, this definition is too broad and is open to alternative interpretations, thus to alternative context management models. In addition, this notion of context is restricted to and only considered through the functionality software agents can deliver. Anyhow, Chen and his colleagues abandoned their own definition of context and acknowledged the one provided by Dey[DA00b].

Central to Chen's context framework is the presence of an intelligent context broker[Che04] that accepts context related information from devices and agents. The presence of this broker is motivated by the need to support small devices that have limited resources for context acquisition and reasoning[CFJ04b].

In spite of them making significant progress on intelligent software agents applications, their real interest, as is the case with other authors[DA00b][Str03] is not context itself, but the way context can be used to enhance the functionality of software agent-based systems. Everything is subdued to this vision to an extent that it not only highly determines the system's functionality, but also determines the relevant components of context they consider into their model.

Interestingly, ontologies actually enhance the functionality of intelligent software agents. Therefore, it is not surprising that Chen was the first author to bring ontologies into the scene in context-awareness to support the Context Broker Architecture (CoBrA)[CFJ04a] [CFJ04b] [CPC+04]. Even more, it may be for this reason that he argued that ontologies are a key feature for the realisation of the Ubiquitous Computing paradigm[CF03][CFJ05]. The CoBrA architecture supports the modelling and reasoning upon context information. He uses the OWL[W3C04b] ontology language to model the CoBrA ontology [CFJ04a] to support (context) reasoning and context knowledge sharing.

Having on the one hand set the path to standard context models by use of ontologies, unfortunately, most of his applications[CFJ04c] were not sufficiently generic and only worked with certain devices (SonyEricsson T68i[1],

[1]http://www.sonyericsson.com/cws/support/phones/t68i?cc=eslc=es

SonyEricsson T800[2], Palm Tungsten T[3]) and at certain (smart) spaces. This, again, is another example of misuse of the term ubiquitous computing.

Some important technical features of Chen's and colleagues' applications is user privacy. Actually, the later version of the CoBrA-ONT[CPC+04], SOUPA[CFJ05], has modelled a so called SOUPA policy ontology, called Rei[KFJ03][Bal07]. Users can define customized policy rules to permit or forbid access to their privative information. The context broker uses a policy-reasoning algorithm that exploits the description logic inference over OWL.

Other examples of semantic-based context models are shown in the work carried out by Gu and Wang in the so called Service Oriented Context-Aware Middlweare (SOCAM) project[GPZ94] [GWPZ04] [GPZ05] [WZGP04]. It is an architecture to build and prototype context-aware mobile services in smart environments[Bal07] and constitutes yet another attempt to set an adequate infrastructure support for building context aware applications[GWPZ04], which in their opinion have to share a common context model. As well as Chen, their context model is based on OWL.

SOCAM uses a central server taht gains context data through distributed context providers and offers it in a relatively processed way to its clients[GPZ94]. They refer to Dey[GPZ05] when defining their notion of context and they further categorize context in person (name, role), location (coordinate, temperature), computational entity (device, network, application) and activity (scheduled activity, agenda)[GPZ05], providing some more clarity to the original definition.

The basic concept of their context model is based on an ontology that provides shared vocabulary to describe specific situations in a domain. It also includes machine interpretable definitions of basic concepts in the domain and relations among them. Modelling context ontologies has been proved to be a very difficult task, for the notion of context is very broad and its conception by each author is different[GPZ05]. Digging into the various ontological approaches provided in the literature, what is interesting about Gu's approach (which is completely shared by Wang[WZGP04], in an almost literal context-work copy) is his (their) sub-division of the notion of context into a more general (high level ontology) and a more particular case of context represented as a domain ontology that can be aligned with the upper ontology[GPZ05].

The upper ontology is a high level ontology, which captures general context knowledge about the physical world. The domain-specific ontologies

[2] http://www.sonyericsson.com/cws/products/mobilephones/overview/k800i?lc=escc=es

[3] http://kb.palm.com/wps/portal/kb/emea/tungsten/t/unlockedeu/home/page$_e s.html$

are a collection of low-level ontologies, which define details of general concepts and their properties in each sub-domain.

The low-level ontology in each sub-domain can be dynamically plugged into and unplugged from the upper ontology when the environment is changed, for example, when a user leaves his home to drive a car, the home-domain ontology can be automatically unplugged from the system; and the vehicle-domain ontology can be plugged into the system.

Tables 2.3, 2.4 and 2.5 show a summary of the most relevant authors in context-aware research that belong to the two generations that have been analysed. It also introduces the definition of context that will be put forward in Section 3.1.2.

Author	Definition	Approach	Context Model	Technology	Assumptions	Applications	Restrictions
Want	Location	Locate individuals in office environment	Abstraction of location	IR sensors	N/A	Forwarding phone calls	Definition, sensors, model
Schilit and Theimer	Where, who and who you are with. Resources nearby	HCI: Application environment	Object Oriented. Sensors and OO	IR	N/A	ParcTab	Def. of context and sensors
Dey	Any information to characterize the situation of an entity, considered relevant to the interaction	HCI	OO	Sensors, OO	GUI Widget Abstractions	Context Toolkit	Sensors, no-semantics

Table 2.3: Table of context authors (I)

Author	Definition	Approach	Context Model	Technology	Assumptions	Applications	Restrictions
Strang	Set of all the information relevant for a task	Distributed Systems	Ontologies	Sensors, Java, Semantics	N/A	Data Interchange at context level	Definition, context model
Chen	Dey	Intelligent SW Agents	Ontologies	Java, Semantics	Agent functionality	Smart Mtg Room	Theoretical support, definition, align with SW agent functionality
Gu, Wang	Dey	Service interoperability	Ontology based, core-domain	Semantics, Java	N/A	Smart Environments	Sensors, lack of theoretical work
Brown	Elements in individuals' environment	Info exchange	XML	XML, Java	N/A	N/A	N/A

Table 2.4: Table of context authors (II)

Author	Definition	Approach	Context Model	Technology	Assumptions	Applications	Restrictions
Ryan	User location, environment, identity, date and time	N/A	XML	N/A	N/A	PDA app.	Sensors
Pascoe	User location, environment, identity, date and time	HCI	Hierarchical	N/A	N/A	Stick-e notes	Sensors
Schmidt	Knowledge about user's, device's state, surroundings, situation and location	NLP, HCI	XML	XML, Java	N/A	Interface dev.	Sensors
Lamsfus et al.	**Information about the visitor, environment and tourism objects**	**Human Mobility**	**Semantics**	**Java**	**Web info. available**	**Concert**	**Context information**

Table 2.5: Table of context authors (III)

2.5 Analysis of Context Ontologies

This section will look in detail at the existing context ontologies, some of which have already been mentioned in the previous section, thoroughly analyzing the concepts they comprehend in order to model context. There are several examples of context ontologies in the literature, none of which have been regarded as *de facto* standard or have been developed by competence bodies such as W3C[4].

One of the first examples of context ontologies is the CoBrA (Context Broker Architecture) Ontology (COBRA-ONT)[CFJ04b] (Fig.2.5), which supports context-aware systems in pervasive environments. It is expressed in OWL[W3C04b] and it represents a collection of terms describing places, agents, events, and their associated properties[CFJ04b]. Ontologies play a central role within the CoBrA architecture[Che03], since they help the context broker share contextual knowledge with other agents enabling reasoning about context. Basically, what the COBRA-ONT does is to enhance the functionality of an intelligent SW and it represents contextual information to the extent that is valid for software agents.

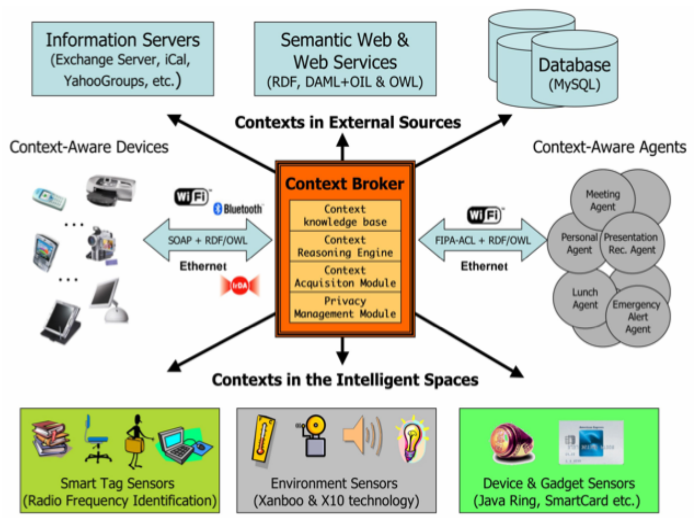

Figure 2.5: The CoBrA architecture

The SOUPA (Standard Ontology for Ubiquitous and Pervasive Computing) was proposed by the same authors that put forward the

[4]http://www.w3c.org

COBRA-ONT[CFJ05]. It represents a shared ontology that has been designed to model and support pervasive computing applications[CFJ05]. It is expressed in OWL and includes modular vocabularies to represent intelligent agents with associated beliefs, desires and intentions, time, space, events, user profiles, actions and policies for security and privacy. It is based upon other ontologies such as FOAF[BM08], DAML-Time[Hob02], OpenCyc Spatial Ontologies[Len95], CoBrA-ONT[CFJ04b], Regional Connection Calculus[LCG08] and the Rei policy ontology[KFJ03]. However, the ontology only poorly supports mobile services and applications[VSS$^+$07]

Very much like the CONON[WZGP04] ontology (which will be introduced later in this Section), SOUPA (Fig.2.6) is also divided in two different ontologies, namely SOUPA –Core and SOUPA-Extensions[CFJ05]. SOUPA-Core defines a set of classes and entities common to almost all scenarios within pervasive computing, e.g. Person, Agent, Policy, Time, Space and Event. SOUPA Extension ontologies extend from the Core and define additional vocabularies for supporting specific types of applications and provide examples for the future ontology extensions.

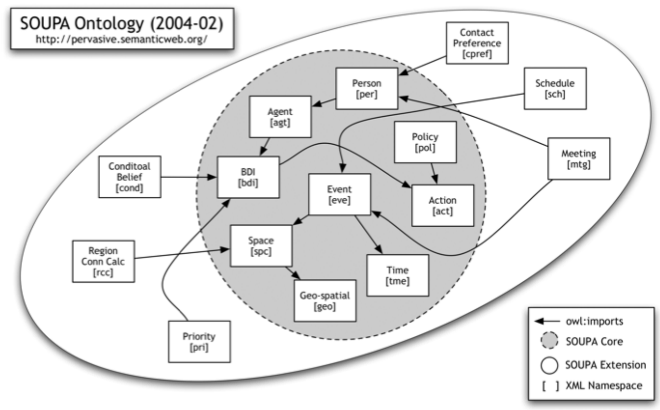

Figure 2.6: The SOUPA ontology

Ontologies have been used in context-aware environments for the sake of supporting knowledge sharing and re-use, logical inference and model consistency check[WZGP04]. One example is the CONtext ONtology (CONON) (Fig.2.7), which models context in pervasive environments. It follows a double ontology approach: there is a set of common classes in a so called upper-ontology that includes general context aspects. Each of the specific domains (represented by a lower domain ontology) would then align to the upper-ontology to support particular context-aware services.

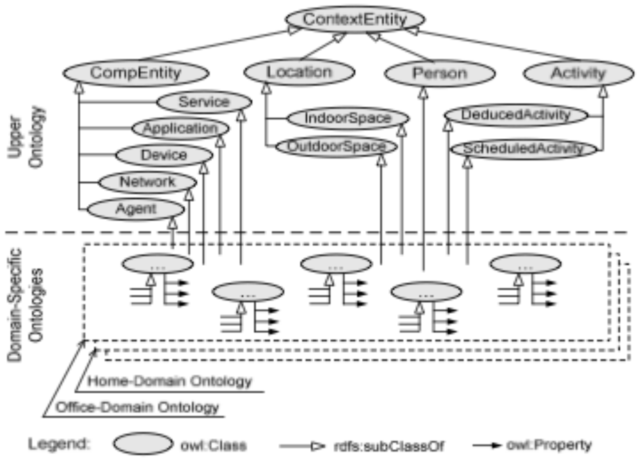

Figure 2.7: The Context Ontology

The general concepts that have been taken into account in the upper-ontology are Location, Person, Activity and Compentity (Computing Entity). This ontology can undertake reasoning processes in order to check the ontology's consistency and to infer higher level context or situations through rule-based reasoning.

Both SOUPA and CONON have been created following the same strategy, i.e. one core and other domain ontologies. However, they comprehend different context concepts, since the objectives of the projects in which they were created were different and therefore, had different requirements of context information.

It is important to note that the design of CONON takes into account three important issues that must be considered when using ontologies, not only in context-aware applications, but also in any kind of applications. Context reasoning has a heavy computational load and if processing time is too high, reasoning may not be undertaken in time-critical applications. However, if the ontology is divided into two parts the computational overload decreases, since reasoning would be performed only in part of the ontology. The other aspect, also implicitly tackled by using ontologies at another level, has to do with the separation of context gathering and context processing and managing. Since important interoperability restrictions derive from the use of sensors, separating these two processes enhances the overall interoperability of the context model, as stated by Dey et al. in their Context Toolkit. Last but not least, it considers scalability as an important aspect of the ontology that allows the upper-ontology to be extensible according to applications particular requirements. CONON is based upon another existing ontology: OpenCyc.

A middleware architecture to support has been proposed by means of an ontology that follows exactly the same conception as CONON, i.e. a double ontology approach in terms of an upper ontology and additional domain ontologies[GWPZ04]. Figure2.8 shows the SOCAM ontology, which actually comprises similar terms to CONON.

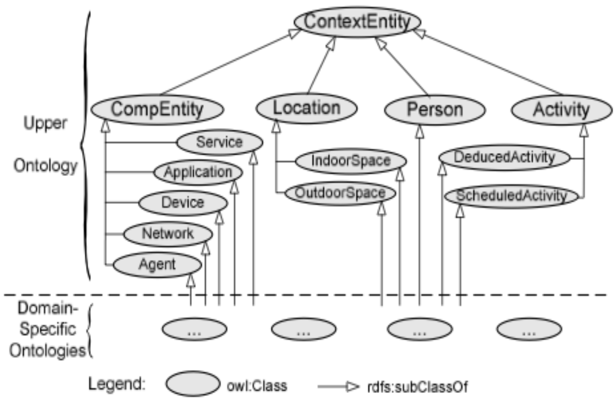

Figure 2.8: The Service Oriented Context-Aware Service Middleware

The Context Management Ontology (COMANTO)[SRA06] (Fig.2.9) defines the main components for context management, as well as the relations among them, i.e. how to gather, organize, process and distribute contextual information. The context management ontologies previously put forward are too specific and COMANTO aims at incorporating the advantages of previous applications in one single solution. COMANTO's structure allows to separate context management from the applications themselves in an effort to enable context interchange and interoperability among applications.

The following are the main classes defined within the ontology: person (this class gathers information about people and relationships among them),

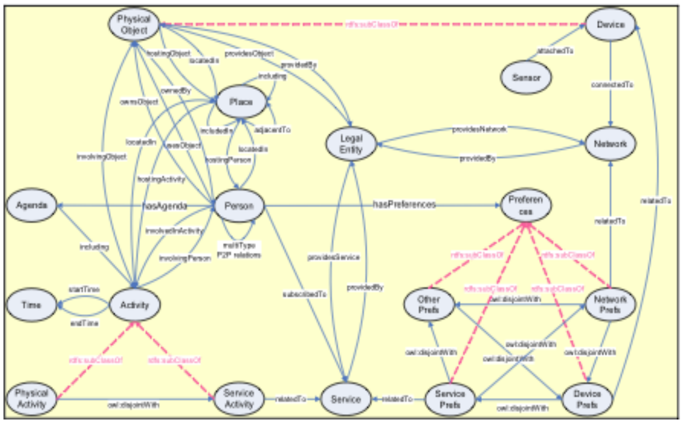

Figure 2.9: The Context Management Ontology

place (this class gathers information about the abstraction of a physical space), preferences (this class gathers information about preferred devices, services, networks and others), agenda (this class gathers information about the calendar in use), activity (this class gathers information about the individual's activity at a particular moment of time), time (this class gathers information about time), physical object (this class gathers information about objects other than devices), sensor (this class gathers information about sources of context information), service (this class gathers information about services and application related information), network (this class gathers information about the network in use) and legal entity (this class gathers information about bodies and/or companies that may take part within the system).

Some of these concepts are also taken into consideration in the CODAMOS ontology[PVR^{+}06], which consists just of four concepts as a result of a requirement analysis done for Ambient Intelligence environments. The concepts are namely, user, environment, platform and service (see figure 2.10.

The GAS ontology[CGK05] was developed in order to semantically describe the basic concepts (objects) within a ubiquitous computing environment. Its main objective was to provide a common vocabulary for heterogeneous devices that constitute a pervasive computing environment to enhance communication and enable more efficient co-operation. This

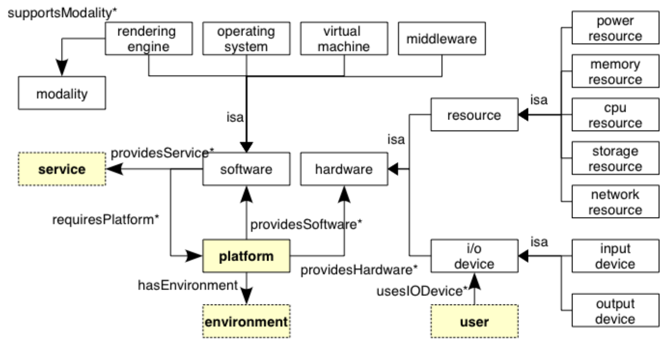

Figure 2.10: The CODAMOS ontology

ontology has been especially designed to fulfil the requirements of service discovery within ubiquitous computing environments.

The ONCOR ontology[KNC07] is basically thought to provide a flexible and practical ontology to describe locations, devices and sensors within ubiquitous computing systems, that delivers personalized information in a building environment. For this reason, the authors have defined an OpenCyc[Len95] and SOUPA[CFJ05] based ontology to describe a building (MIBO). The basic entities that this ontology takes into account are people, spaces, devices and services that can be found inside the building. One main difference and advancement with respect to previous ontologies is that MIBO considers context history in order to enrich reasoning.

The General User Model and Context Ontology, GUMO has been developed to model a user's dimensions[HSB+05]. This ontology is highly influenced by UserML, SUMO and UbisWorld.

Shehzad and colleagues[SNLL05] defined a context-aware system in which contextual information is modelled through an ontology. They created different categories in order to manage the heterogeneity of entities that they take into account within the context. The categories are the following: Agent, Environment, Device, Location and Time.

There are other ontologies in the literature that have not been thought to manage context as whole, but different of its aspects and constituents. The Generic User Profile defines a user profile structure[5].

The Friend of a friend project, FOAF, is based in the usage of machine processable and understandable personal, companies Web pages[BM08].

The SPICE Mobile Ontology (Service Platform for Innovative Communication Environment) focuses on the design, development and implementation of a service creation and execution in advanced communication networks, such as 3G[VSS+07].

The Composite Capability/ Preference Profiles (CC/PP)[W3C04a] is a description of device capabilities and user preferences. The W3C Ubiquitous Applications Working Group continued working on the CC/PP initiative and developed the so called Delivery Context Ontology[W3C08]. The Delivery Context Ontology represents a forma model of the characteristics of the environment in which devices interact with the Web or other services. They delivery context includes the characteristics of the device, the software used to access the service and the network providing the connection among others. The ontology is formally specified in OWL[W3C04b].

As can be seen from the previous review on context ontology, none of the ontologies that have been analysed considers the characteristics of human mobility as they will be explained in Chapter 3. Some of the ontologies consider concepts that may be of interest and that may be partly re-used, however, none of the ontologies, for example, models the human being with its characteristics when *en route*. The lack of addressing this and other issues make it necessary to design a new context ontology (Chapter 4) based on the theory development (Chapter 3). A summary of this review is shown in tables 2.6 and 2.7.

Criteria have been establish to objectively compare the ontologies, including ontology specification language, the concepts that the ontology comprehends and the type of software it would support when used in an application environment. These are the base criteria for the framework of this dissertation and that will be taken into account in Chapter 4 in the ontology building process, in order to decide which ontology or part of ontology shall be re-used.

[5] http://www.3gpp.org/FTP/Specs/html-info/29240.htm

Name	Spec. Language	Approach	Applications	Concepts	Type of Software
COBRA-ONT	OWL DL	Intelligent SW Agents	Smart meeting rooms	Places, agents, events, properties	Java, mobile devices
SOUPA	OWL DL	Intelligent SW Agents, CoBrA-ONT. Upper and domain	Smart meeting Rooms	COBRA-ONT	Java, Mobile devices
CONON	OWL DL	Double ontology: Core - domain	Smart (indoor) environments	Location, Person, Activity and Compentity	Type of software
SOCAM	OWL DL	CONON	Smart (indoor) environments	Location, Person, Activity and Compentity	Type of software
COMANTO	OWL DL	Exchange context informa-tion	Mddlware for context info.	Person, Place, Preferences, Agenda, Activity, time, Physical Object, sensor, service, network, legal entity	Type of software

Table 2.6: Context Ontologies (I)

Name	Spec. Language	Approach	Applications	Concepts	Type of Software
CODAMOS	OWL DL	Ambient Intelligence	Smart Environments	User, environment, platform and service	Java, mobile devices
GAS	OWL DL	Object description in semantic environments	Vocabulary for heterogeneous devices. Service discovery	Terms	Type of SW
ONCOR	OWL DL	Location, device, sensor	Ubiquitous Computing	Location, devices, sensors	Information Del.
SPICE	OWL DL		Smart (indoor) environments	Location, Person, Activity, Compentity	NGN Service Del.
CC/PP	OWL DL	Device capabilities and user preferences	APPS	Characteristics of device, software used to access the service, network	Type of software

Table 2.7: Context Ontologies (II)

2.6 Tourism Mobile Applications

Most of the context-aware theory, systems and applications have been developed with the primary objective to support humans' interaction with mobile computing devices in various ways. The examples of the applications that have been reviewed so far in this Chapter have not been implemented in the tourism domain, possibly due to the fact that the amount of assumptions and preconditions of those applications present serious barriers for the requirements of mobile tourism guides.

However, mobile guides for tourism have a long tradition[GPW+08] since the first prototype was built back in the mid 90s[AAH+97]. As mentioned in Chapter 1, tourism is a well suited application domain for location and context-aware type of applications, for tourists can be assisted not only in the pre-trip phase of a journey but especially during the journey itself[SGPR09] significantly enhancing their tourism experience.

This Section presents some of the most relevant mobile tourism guide and analyses them according to the criteria established in the following Subsection.

2.6.1 Evaluation Criteria

In order to evaluate and be able to more clearly extract some conclusions from the following review, the next list shows the evaluation criteria that have been established to categorize existing tourism mobile guides:

- *Location-based*: This refers to whether a tourism mobile application uses some location information.
- *Context-aware*: This refers to whether a tourism mobile application uses, apart from location information, some other information such as user data, user preferences, environmental information and so forth.
- *Technology*: This refers to the communication technology used by mobile tourism applications to access to Web-based services.
- *Semantics*: This refers to whether the tourism mobile guide makes any use of semantic-based technologies for any purpose.

2.6.2 Examples of Tourism Mobile Guides

Several mobile tourism guide surveys have already been published[EOSB03] [KB03] [BCK05] [GPW+08] [SGPR09]. The first three reviews primarily analyse early generation mobile guides and investigate specific issues, such as support for maps or mobility aspects. However, they do often not provide a comprehensive insight into the kinds of services that are supported and moreover, in depth technical aspects of how these services are delivered to the end-user are also to be missed[GPW+08].

The massive social adoption of advanced multimedia-based mobile devices, enhanced with GPS features and equipped with a variety of communication technology systems have resulted in a large variety of mobile service ranges. Thus, this review mostly focuses only in last generation mobile tourism applications[GPW+08][SGPR09] which are more similar in features and characteristics to the application developed with base upon the contextual computing framework put forward in this dissertation than early prototypes[AAH+97] [CDM+00] [PBHS03] [HB06] [SGK05].

The most important conclusions that can be drawn from Grün's and colleagues' evaluation is that most of the mobile tourism guides either provide location-based information services or concentrate on delivering personalized information, i.e. they fail to provide a combination of both as a rudimentary context-aware kind of service.

One major drawback of the systems that have been reviewed, both first generation and second generation mobile tourism guides, is the amount of (active) interaction they require form an end-user point of view.

There are just two examples real context-aware mobile tourism guides[Hoe] [HFZ+06] [BFH+07] [HÖ8] [Bee09]. Interestingly, although the CAIPS system[BFH+07] provides rule-based push information, it fails to provide a general framework to support human mobility, since it is more focused on creation of rules for information delivery[Bee09].

Table 2.8 shows a summary of the most relevant tourism mobile applications.

Name	Device Type	Technology	Comm. Tech.	Nature	Context	Restrictions
Cyberguide	Hand held	N/A	TCP/IP	Pull	Location (IR and GPS) and orientation	Sensors and amount of interaction
GUIDE	Tablet PC and hand held	N/A	Cell-based comm., Wave-LAN (IEEE 802.11)	Pull	Personal and Environmental context	Sensors and interaction
CATIS	Hand held	Java, .NET, XML, HTML	HTTP	Pull	Location, time and preferences	Sensors and interaction
TIP	Hand Held	Java	N/A	Pull	Location, preferences and travel history	Sensors and interaction
MyCampus	Hand held	N/A	Wi-Fi	Pull	calendar, location, user preferences, organizational databases	Sensors and interaction
CAIPS	Mobile device	Java, semantics	Wi-Fi, 3G	Push	Dey	Pre-conditions, framework
etPanner	Mobile Device	Java, semantics	3G	Pull	Dey	Pre-conditions, framework, sensors

Table 2.8: Tourism Mobile Applications

2.7 Summary

As can be seen from this Chapter, the amount of research conducted in context theory is vast and the number of applications developed is almost countless. However, it is a fact that context-aware applications are not available to the general public. Why? Why do context-aware applications only exist as prototypes or in certain laboratory environments? Why do context-aware applications not exist in tourism? This section shows the summary and conclusions extracted from the review done aiming at answering these questions. These conclusions can be added to the list of motivations shown in Chapter 1, as extra arguments that support this dissertation.

In first place, it is important to note that theoretically and conceptually speaking a sufficiently agreed definition of the notion of context does not exist within the scientific community. Therefore, neither does an agreement exist on how contextual information ought to be managed. This is due to various reasons: Researchers have approached the notion of context from various distinct backgrounds, as mentioned in other Sections of this Chapter, namely human computer interaction[DA00b], intelligent software agents[CFJ04b] and distributed systems[Str03] setting already a natural precondition to the notion of context that will result from their research work. Besides, the objectives researchers pursued were linked to their own background expertise and not to the study of context itself as a research discipline. Furthermore, the existing definitions had some room for interpretation. Thus, the particular interpretation each of the authors made on the definitions considering their background expertise together with their objectives, lead to different context management methods or techniques and context constituents. In addition to the models of context being qualitatively different, they are also quantitatively different, i.e. each context model manages different types of information, since context modelling requirements within each discipline from which context has been approached are not necessarily coincident. The different context models originated from the various interpretations have produced very particular and *ad hoc* applications, which make their re-use in other environments hardly possible.

Thus, there is a need to more deeply study the nature of context in the different research disciplines and establish valid requirements for context modelling in each of them. In other words, context is contextual and highly dependant on the framework in which context theory is going to be implemented. What is valid in (the notion of context for) human computer interaction does not necessarily have to be valid in (the notion of context for) intelligent software agent research or in (the notion of context for) the study of human mobility. Moreover, this has direct implications on the requirements for context modelling, which have to be reviewed, defined and set for each case (see Chapter 3).

Existing approaches to context-awareness in the framework of human mobility (reported in previous sections) have literally imported the conception of context put forward in a totally distinct discipline, i.e. human computer interaction without really digging deep into the nature itself of human mobility ignoring the requirements for context modelling within the framework of human mobility. Not surprisingly, the consequent models do not completely suit the requirements.

This is one of the major problems in the existing work applied to tourism and to other disciplines. Context (in a human mobility framework) is sufficiently particular and distinct to context in other disciplines as to conduct research that originates new knowledge derived from the observation of the nature of human mobility. This knowledge will produce a new definition and a new context information management model, directly focusing on the problems and questions that arise when tackling context within human mobility, not when tackling other disciplines. This is one of the reasons why a radically different approach to context is presented in this dissertation, which considers the nature of human mobility and establishes the context modelling requirements under this framework. The framework proposed in this dissertation aims at exploiting the notion of context on the benefit of visitors (humans on mobility) and using and developing technology that will enable them to effectively use context-based information. All of this is thoroughly explained in Chapter 3.

All of the applications that have been reviewed throughout the Chapter use sensors to gather contextual information. Although sensors facilitate various distinct kinds of context information (temperature, level of noise, lightning, and so on) using these devices is a cumbersome task: first, a chosen area has to be covered with sensors. Then, the entire infrastructure needed to (physically) transport sensor raw data to a machine has to be built and finally, sensor raw data has to be converted into machine processable code (different in each platform) to make some sense of it. Furthermore, sensors built by different manufactures use different programming languages and they measure in different units. All of this (just) adds more complexity to an already complex problem (on its own), which is building contextual computing applications. In addition, it restricts the use of context-aware applications to the locations that have previously been populated with sensors, which has created major inconsistencies in the use of some words: the authors have built context-aware systems for pervasive computing environments, however, all of these systems only run in certain pre-defined environments, with certain pre-established mobile devices, under certain very concrete circumstances which is the contrary to what Weiser had as a vision[Wei99].

There is one second issue in context-aware research that does not only imply significant interoperability pre-conditions and increases the amount of work to carry out, but also poses serious barriers for context sharing, re-use and integration: not having a standard context modelling and managing method.

In an effort to achieve interoperability of context-aware systems, two authors proposed to separate context gathering from context information management[Dey01] [SRA06]. Dey's and colleagues' Context Toolkit relies on the concept of widgets. Just as GUI widgets mediate between an application and the user, context widgets mediate between the application and its operating environment hiding sensing low level details. However, the widgets have not shown to have a robust behaviour. In addition, all sensors have failure modes that make the data they send to the application not (100%) reliable[DAS99]. The context-widget based architecture does not completely support the transparent data acquisition of context for applications. In order for an application to use a widget, server, or interpreter it must know both the hostname and port the component is operating on. If a sensor fails to work, all the communication system has to be reconfigured. In the same way, the structure of COMANTO[SRA06] allows to separate context management from the applications themselves in order to enhance context interoperability and interchange among applications.

Based on the previous approach, other authors[CFJ04b] [GWPZ04] [WZGP04] [SRA06] [PdBW^{+}06] have used in addition ontology-based context models to foster interoperability at the data level. However, their models' dependability on sensors poses serious restrictions for their re-use outside the environments for which they have been designed. The same is true for Chen's Smart Spaces applications [CFJ04c]. They need to populate one particular room with sensors and their infrastructure in order to be able to gather (contextual) information from that room. This is a major disadvantage for this and other applications[Wan92] [SAW94] [Dey00] [CFJ04b] [GWPZ04] [WZGP04], not because of the major technical problems and handwork overload it creates, but even more, for it restricts the use of these kinds of systems to areas that have previously been populated with sensors.

One major issue nowadays in context-awareness is user privacy. Some of the authors [Wan92] [Dey00] [CF03] [HB06] have tackled it in one way or another aiming at protecting personal information. In spite of their work, they all recognize that there is still much to be done.

All in all, the applications that have been reviewed in the literature imply a high degree of human intervention both at a development phase and at a user intervention phase, as for example Chen's and Dey's applications. On the one hand, all the sensors have to be placed somewhere and all the (wires, mostly) infrastructure has to be built. All of the sensor information has to be translated into something meaningful for the supporting architecture. On the other hand, users need to very intensely and explicitly interact in order to have their (mobile) devices ready to run in the smart environments. In addition, users are required to use certain mobile devices.

Most of these applications have been designed to be used in indoor environments, except for the tourism mobile guides that work also in outdoor environments. The restriction in the case of indoor applications comes once again from the fact that specific sensors and devices are needed in order to make the applications work. In the case of the tourism mobile guides, despite the fact that can be used in outdoor environments, they have various kinds of restrictions: firstly, they have not an insight of the notion of context within human mobility and the notion of context they present is very limited to really support humans on the move. The devices they use have low computing capabilities and most of them are out of use, such as the so called PDAs or handhelds. These devices are also dependant on the communication technology to be used. Most of them use 3G communication technologies, which as said in the introduction, happens to be too expensive when under roaming conditions.

None of the aforementioned context-aware systems (tourism guides or any other kind) use digital broadcasting technology. This technology is however extensively used for traffic information applications, such as Traffic Message Channel (TMC), or applications proposed by the Transport Protocol Experts Group (TPEG) (http://www.tisa.org). One of the main characteristics of this connectivity technology is its "push" nature, which makes it particularly interesting for the tourism domain[BFH+07] relieving the visitor from having to actively search and browse for information[GW04]. Out of all the applications reviewed, only the CAIPS is push-based.

Chapter 3

CONCERT: Conceptual Foundations

"It's time to get focused..."
Tim Smithers, (1957), English research scientist.

The framework for contextual computing in tourism, named CONCERT, aims at providing tourists with context-based services by reducing the complexity of context-aware application development on the one hand (see Section 1.4), and by making these kinds of systems universally accessible on the other. In order to accomplish these challenging objectives, two fundamental issues are addressed: Firstly, the conceptual approach to the notion of context in CONCERT is radically different to the existing ones. Secondly, CONCERT proposes a double level interoperability schema: the first level of interoperability is provided at the infrastructure level and is achieved by not using sensors to gather contextual information. The second level of interoperability is provided at the model level and follows similar approaches to other existing ones, i.e. using semantic technologies in order to enhance context data interoperability, share and re-use.

Providing interoperability at two levels (Fig.3.1), i.e. at the hardware (infrastructure) level (avoiding sensor and other complex infrastructure) and at the model level (interoperability of data) could set the conceptual foundations for meaningful contributions in the fields of contextual computing in general and in contextual computing applied to tourism in particular towards achieving Mark Weiser's original ubiquitous computing vision[Wei99].

These two characteristics of the CONCERT framework are explained in more detail as follows:

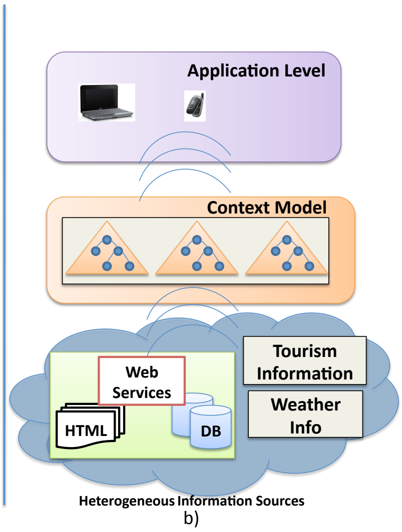

Figure 3.1: The two level interoperability proposal in CONCERT

- *Context theory in the framework of human mobility*: CONCERT's objective is to study the context of visitors in the framework of human mobility in order to more precisely determine the information that formally describes that context, i.e. to establish the requirements in terms of context information. Context is a main entity, not an auxiliary variable for studying something else as in earlier approaches. So, the first thing to do is to find out what is that defines the context of people on the move and what minimum information would be needed in order to define that person's context. Then, the following step is to determine the sources information will be retrieved from, where they are, and how they can be accessed.

- *Two level interoperability schema* (Fig.3.1):

  - *Network of ontologies (interoperability at the model level)*: The notion of context has to be translated into a consistent computing model in order to effectively process context information automatically. Different context models have been identified and analyzed according to ubiquitous computing environments requirements[Str03][PdBW$^+$06] and context-aware applications[CK01][BBH$^+$09]. Both approaches indicate that ontologies clearly fulfil all requirements and are so far one of the most adequate in order to model contextual information. Besides, ontologies have proven to be good intermediation tools in information integration. In addition, ontologies can also provide reasoning functionalities that are valid for the context model to entail implicit context knowledge. This is crucial to achieve the vision of context pursued by the CONCERT framework. In an attempt to increase the level of scalability, modularity and interoperability at the model level, context shall be modelled in CONCERT by means of a network of ontologies[HRW+06].

– *No sensor infrastructure (interoperability at the infrastructure level)*: Populating cities, regions and/or open areas with networks of sensors is not affordable. In order to tackle this barrier, the CONCERT framework gathers (contextual and tourism) data from the Internet as well as from mobile embedded sensors, such as the GPS sensor. Thus, the framework does not require further complex infrastructures and having to populate a particular area of interest with sensors would be avoided on the one hand, and on the other, the use of a CONCERT framework based application would not be limited to those areas. In fact, this is one of the trends nowadays: to publish more and more sensor data on the Internet in the construction process of the Future Internet, and use mobile devices to establish a *dialogue* between the data and the real world[1] . The objective is not to contextualize a particular environment, but to increase the level of abstraction of context (thus, increasing the level of portability -interoperability- of the model), by providing the model with the means to model a particular individual and incorporating the data about location in a later phase through the GPS sensor. The context model becomes independent from location while maintains this information in the model, which as demonstrated in previous chapters is fundamental for context-based applications[GPZ05][Ay07].

These characteristics are the main theoretic contributions and the base upon which the rest of the contributions are grounded. These, together with the notion of context that is suggested in the next Section (3.1), will have a key role in determining CONCERT's future functionality[Str03]. Figure

[1]An example of this would be the *Pachube* initiative, http://www.pachube.com

3.2 shows the flow of implications due to the theoretic approach to context suggested in this dissertation.

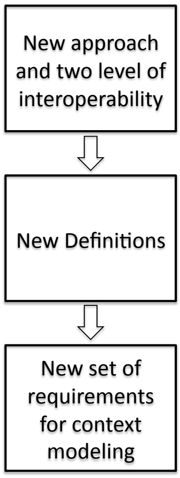

Figure 3.2: CONCERT's flow of theoretic contributions

A direct implication of addressing the issue of context from another perspective is that some terms have to be redefined in order to clearly set the framework. Thus, Section 3.1 presents the definition of the terms, which are relevant to this dissertation.

Other direct implications (arisen from the definition of context) have an impact on context modelling requirements, i.e. information requirements, nature of information, context modelling requirements and context information gathering requirements, as depicted graphically in Figure 3.4. These are specified in detail in Section 3.2.

Following, the use of ontologies to manage context information will be thoroughly justified in Section 3.3 and the Chapter will finalize with some concluding remarks in Section 3.4.

3.1 Definitions

3.1.1 Definition of Contextual Computing in Tourism

Most of the literature refers to context-aware systems and/or applications to denominate systems that make use of information that originates within the context in which they run[SAW94][CK01][Dey01][RJSFG06]. These applications have been programmed to automatically react (in various ways) to changes that occur in their environment without explicit human intervention.

The term *context-awareness* has been given a lot of attention by researchers as well: some definitions have been proposed[SAW94][Dey01] and classification of context-aware services established in terms of categories[CK01]. Thus, the following list presents some of the features context-aware applications ought to have[Dey01]:

- Presentation of information and services to the user.
- Automatic execution of a service.
- Tagging of context for later retrieval.

Considering the above, there are basically two types of context-aware computing[RJSFG06]:

- *Use of context for diverse purposes*: applications that utilize users' context knowledge targeting software automation[Dey01][DA00b]; computing devices that detect, sense and interpret users' local environment[Pas98][Pas01]; and the ability to provide applications maximum flexibility based on real-time context sensing[SDA99]. An example of these kind of applications would be service provisioning.
- *Adapting to Context*: applications that dynamically change or adapt their behaviour based on the context of the applications and the users[Bro98]. E.g. changing the ringing tone of the mobile phone to silent mode in a meeting.

Since CONCERT is intended to support human mobility, it would be considered within the first kind of context-aware computing, i.e. use of context for diverse purposes. The objective is not to be aware of context, but to use it to make some sort of understanding about it and based upon it, automatically process tourism related information. CONCERT continuously receives tourism broadcasted information and filters that information depending on the value context has at a certain moment, just as one could set a certain radio station in a vehicle's radio set.

Mobile devices are given the capacity to automatically process tourism information with data that has been originated within people's context (contextual information) and consequently produce some information that is relevant for people on the move. Relevancy here does not depend on the users' (visitors') task[Dey01], but both on visitors' objectives at that particular moment of time and on the value of context constituents of visitors at that same given time[LASM+09b].

The term computing (or computation) is used throughout this dissertation in an open way, as defined by the Oxford Dictionary of English[2], i.e. *the scientific knowledge and techniques that enable automatic information processing by use of electronic devices.*

Therefore, a new adapted definition for the notion of contextual computing in tourism is suggested as follows (definition 1):

"Contextual Computing is the scientific discipline that studies and observes the context of an individual and pursues to generate knowledge out of that observation in order to establish the context modelling requirements and context information management methodology so that tourism information can be processed in a way that is useful to support people's mobility".

Contextual computing is a key aspect of tourism nowadays. By definition, visitors are people outside their usual environment[UNW08b]. Under these circumstances they have certain needs and requirements different from the ones they have while at their usual location, which can be met with information-based services. As mentioned earlier, tourism is suffering a dramatic transformation: it is no longer a labour-intensive industry; rather, it is evolving into an information-based industry in a large extent, due to the impact Information and Communication Technologies have had on it[Poo03][She97][Buh03]. Society in general and tourists in particular are changing their behavioural patterns and more commonly use electronic mobile devices[BCGB07].

Within the CONCERT framework context (and all the features that define it) are information facilitators, i.e. catalysers, the glue or the bridge between the user and the enormous amount of tourism information that is being offered to them through the Internet. Context is the link between the need for information and the information itself. This idea of context specifies Weiser's conception of "anytime" and "anywhere", i.e. it represents particular pieces of information visitors need or want to access in an automatic manner, and the way that information ought to be displayed on the device's screen.

[2] http://www.askoxford.com/?view=uk

3.1.2 Definition of Context

The literature review in Chapter 2 shows most of the definitions of the notion of context that have been suggested through the history of context-aware computing research. Not only does an agreed definition not exist, but the ones that exist cannot be directly imported into the realm of tourism, for they do not fully meet the requirements of the framework (in terms of information that needs to be modelled) in which they are to be used, as will be seen later in Section 3.2.

The nature of the notion of context is too broad on the one hand, and has been used in too oversimplified manners on the other, depending the perspective with which it has been attempted. Therefore, just a single (officially established) definition of context would very likely not span neither the whole meaning of the notion itself, nor would be valid for all different domains. As stated in the the summary of Chapter 2, context is contextual and it needs to be defined for each discipline. In addition, the definition of context needs to be as concise, consistent and precise as to derive an implementable context information management model out of it, which is not always possible due to both the complexity of contextual computing applications and to the lengthy definitions of context put forward, which are not optimal from an operative point of view.

It was first Gu and his colleagues[GPZ94][WZGP04] and then Chen[CFJ05] who suggested a practical subdivision of the notion of context at the implementation level in what they called *core* (context) ontology and *domain* ontologies, in an attempt to boost an easier context management model. This is a convenient approach to improve systems' interoperability, which is enhanced with the use of ontologies at the model level. However, those propositions lack a solid theoretical grounding, since they are not supported by an analysis of the context information requirements for each application. Based on this subdivision in an attempt to shed some light, the definition of context introduced in this dissertation considers different categories of information.

Given the broad scope of the concept of context, Dey suggests that (in the most abstract level) context is *any information that characterizes an entity*[DA00b]. His definition continues by adding various restrictions, which are intrinsic to his background expertise and research interests and objectives, namely human computer interaction. Following the same logic, context within this dissertation is also considered as any information that characterizes the situation of an entity. However, for the purpose of this work the entity is univocally specified: the entity is a visitor[UNW08b].

As well as Dey's definition continues by adding restrictions (valid for context within the framework of human computer interaction), the definition proposed in this dissertation needs to be added other restrictions that help set its scope within the framework of human mobility. These restrictions ought to lead to an own definition of context in the realm of human mobility, clarifying context information requirements. Two different techniques have been used in order to address this matter, i.e. find out the constituents of context in the framework of human mobility:

1. *Brainstorming sessions*: The brainstorming sessions have been held in order to: Firstly, elucidate the maximum number of context constituents. Secondly, sort them out according to the needs in the framework of human mobility. The results can be seen in figure 3.3.

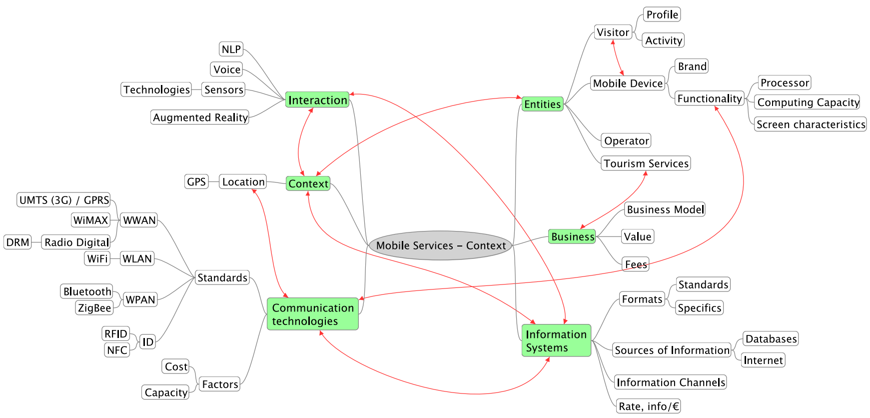

Figure 3.3: Process of brainstorming to define the notion of Context

2. *System Operator*: The System Operator (or 9 Windows) is an analysis methodology followed in the TRIZ innovation methodology[3]. This tool is primarily used for problem analysis and supports people to think in terms of time and space. The basic principle that underlies this tool is to divide the world in nine segments, as shown in table 3.1.2. The central box (system, present) is the one humans' brain naturally migrate to whenever they are given a problem[Man02]. The results of the analysis process performed by use of the System Operator on the notion of context can be seen in table 3.1.2.

[3]http://www.triz.org

	Past	Present	Future
Super System	Components, Mobile dev. char.	Task, activity, interaction	Future context, recharge points
System		Visitor	Agent Symbiosis, Information update, Recharge of battery, new ways of carrying device
Sub-System	Mobile phone technology, visitors' culture, Internet, Smart Tags, Motivation, Needs	Mobile phone components, Visitors' demographics, delivery of information	FET, evolution of mobile phone characteristics

Table 3.1: The TRIZ System Operator or 9 Windows

Being however Dey's definition the most referenced[DA00b] [Bal07] [BBH$^+$09], this dissertation proposes a new definition of the notion of context (based on Dey's) adapted to the requirements and needs of human mobility (elucidated through the previous two tools) in order to establish an operative definition and its scope. This definition is not intended to cover the notion of context in a general way, but just to set up a new paradigm for contextual computing based services in tourism in order to support human mobility, as well as an operational and integral definition. Thus, context within the framework of human mobility is defined as follows (definition 2)[LASM$^+$09a]:

Context is any relevant information that characterizes the situation of a visitor. A visitor is a human outside his usual environment and his situation is specified by data that concerns a)the individual itself and its goals at a given moment of time, b) the individual's environment and c)the tourism objects identified within that environment which can be offered to the visitor.

In order to more consistently and univocally elucidate the definition of context, it can further be specified by enumerating the different constituents that form each category proposed within the definition of context:

- *Data about the individual itself*: this category refers to who the individual is, profile (preferences, demographics), motivation, role as defined by the UNWTO[UNW08b]. This category also contains information about the mobile device the individual is carrying and about the motivation and planned behaviour of the individual activities the individual is going to undertake next.

- *Data about the environment*: This category refers to data concerning location, weather conditions, network connectivity and time.
- *Data about the tourism objects*: This category refers to tourism data that can potentially be relevant to visitors given their location and that needs to further be processed according to the rest of the values of context.

3.1.3 Definition of ontology

Central to the conception of the CONCERT framework is the use of ontologies to implement the context model, as will be described in Chapter 4. In particular, as stated at the beginning of this Chapter, a network of ontologies shall be used for that purpose. The literature review in Chapter 2 has shown various definitions of ontology and has analyzed the concept from its original philosophical conception to today's most extended definitions going through its AI notion in computer science.

However, given that ontologies are tools used in KR, it is necessary to make the following clear differentiation among data, information and knowledge. Data is a set of discrete, objective facts about events[DP98]. Information could be described as a message, i.e. there is an emitter, transmitter and a channel for information to be disseminated from the first one to the second one[DP98]. Finally, knowledge in the framework of this dissertation is the set of properties and relations between facts and information that allow to infer implicit information. Table 3.2 summarizes these concepts.

Ontologies are tools to represent (any kind of) knowledge[NFF^{+}91] by means of a representation language. These representation languages have their own syntax and structure, which is based on the hierarchical classification of concepts in a kind of taxonomy, their properties and relations. Given that the notion of ontology is relatively abstract on its own, in an attempt to shed some clarity on its meaning, the following definition based on the syntactic structure of an ontology is proposed as follows (definition 3):

An ontology is a set of formally defined concepts, attributes of these concepts and the allowed relationships and restrictions among them that

Item	**Description**	**Example**
Data	Discrete, objective fact	A number, e.g. 5ºC
Information	Emitter, receiver, communication channel	The temperature outside is 5º C
Knowledge	Relation between data, information	It is 5 degrees, therefore it is cold outside

Table 3.2: Example of data, information and knowledge

characterize a particular domain of interest. The formality and expressivity of an ontology are given by its representation language.

The domain of interest within this dissertation is the context of a visitor, which is going to be modelled by the use of formally (formality in this case comes from the nature of ontology representation languages proposed by the W3C) defined concepts, attributes and allowed (through restrictions) relationships.

So, considering the previously suggested definitions, the following shows the understanding of the notion of *context-ontology* as used throughout the dissertation (definition 4):

In the travel and tourism domain, a context ontology is the formal representation of a portion of reality of a traveller at a particular moment of time by means of an ontology representation language.

3.2 Context modelling requirements

The origin of pervasive (or ubiquitous) computing and context-awareness is not to be found in tourism sciences, which have traditionally been closely related to social sciences far away from information and communication technologies, but in the realm of human computer interaction and, at a lower extent, artificial intelligence. However, as stated earlier, location-based and personalized information services and applications have widely been implemented in the last years in the realm of tourism. In spite of this, these applications have been developed based upon a theory that has literally been imported from other disciplines, as stated earlier, i.e. without really studying the nature of human mobility and its characteristics, thus not allowing tourism mobile applications to incorporate essential information intrinsic to human mobility into the context model.

This Chapter has begun by pointing out the main characteristics of the CONCERT framework and following, it has suggested a number of definitions. It will now dig into analyzing the characteristics of human mobility in order to establish the requirements in terms of context information that need to be taken into consideration in the context model. In this sense, the questions that the set of requirements needs to answer are the following:

- What type of information does the model need to process?
- What characteristics does that information have?
- How should that information be put together to form a model and in order to be processed in a way that the objectives put forward earlier for the CONCERT framework are achieved?
- How is that information disseminated throughout the model?

Figure 3.4: Categories of context information requirements

Each of the questions of the previous set has derived in different categories of requirements shown in Figure3.4: (i) requirements in terms of information about context constituents, (ii) requirements in terms of information nature, (iii) requirements in terms of the kind of context model to be implemented and (iv) requirements in terms of communication technology to be used to disseminate context information. Thus, in the following sections, each of these requirements will be analyzed in detail. Table 3.2 shows schematically the relation among the different requirements.

Set of Requirements	**Related to requirement**
Context Constituents	Information nature, Context model, Information gathering
Nature of context information	Context model, Information gathering
Context Model	Nature of context information
Information gathering	Context model, context constituents

Table 3.3: Relation among the set of Context Information Requirements

3.2.1 Requirements in terms of constituents of context in human mobility

The definition of the notion of context together with the context model have been identified to play a crucial role in determining the system's future functionality[SLP04][KS07]. In fact, the new definition of context on its own poses a number of requirements in terms of the type of information that needs to be comprehended in order the model to be coherent with the theory that supports it.

Picture 3.5 shows the different information categories identified within the definition of context suggested earlier in this Chapter with the variables that comprise them.

This information has been obtained as specified in Section 3.1.2. The concepts located on the right represent the requirements of information that the model needs to cover in terms of type of information, i.e. what information is needed to gather. However, is this information what really characterizes the context within the framework of human mobility? What is the essence of the context within human mobility? What is different in this context from other contexts? What information defines better a human

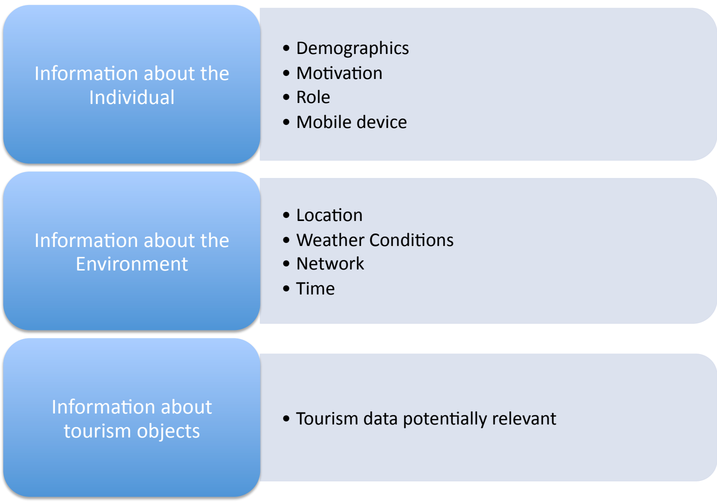

Figure 3.5: Categories of context suggested by the new definition

mobility environment? The essence of mobility here is represented by the visitor, i.e. the information that characterizes the situation of a visitor is key in order to be able to precisely model context within the framework of human mobility.

In order to shed some light in this discussion, the term *visitor* and the features that characterize it have already been specified by the UNWTO[UNW08a]. These are the characteristics that shall be considered and incorporated into the context model (Chapter 4) in order to take into account the real needs of the context model in terms of visitor information.

The definition of context has implications not only in the overall functionality the framework can provide, but also in the rest of the requirements identified, i.e. nature of information in human mobility, context information modelling and management and context information gathering as shown following:

- *Implications for information nature*: All the terms that define each category are susceptible to have the characteristics shown in Section 3.2.2, Requirements in terms of the nature of information in human mobility.
- *Implications for the context model*: Context information has to be modelled in a way that it can be re-used, shared, and interchanged

across different contextual computing applications. That to happen, context information needs to be machine processable. In addition, the power of the model will be partly dependant on the relations that are established amongst the terms themselves.

- *Implications for information gathering*: Context information may be formatted in differently depending on where and how the information is retrieved and disseminated.

It also has major implications in the design of the interface. As will be shown later in Section 5.4.4, visitors themselves need to provide some information to the system in order to complete the context information the system needs to filter tourism data. Therefore, the information needs to be easily provided into the system not to discourage visitors' use.

3.2.2 Requirements in terms of the nature of information in human mobility

Any contextual computing system is necessarily a composition of distributed data and information that lacks of a central entity for its management[SLP04]. Contextual information is highly dynamic in terms of its variation in time, network, source, availability and quality, amongst other features and its nature has high implications on the kind of model that better suits these requirements. The most remarkable requirements and features of context information, also valid for CONCERT, have been summarized as follows[CK01][SLP04][KS07]:

- *Distributed Composition*: This is perhaps one of the most representative characteristics of context-aware systems: context data is distributed (sensors are placed at different locations) and heterogeneous (the nature of data is different from sensor to sensor). There is no central instance that is responsible for the creation, deployment and maintenance of these data. This will be the main characteristic in the CONCERT framework as well: context information has different sources, e.g. different Web pages and services, visitor provided information and mobile device embedded information (distributed) and different nature (heterogeneous) and has to be integrated for the whole system to work coherently.

- *Partial Validation*: It is needed to (at least) partially validate contextual information on structure as well as on instance level against the context model in use even if contextual information is not available at a certain moment due to its distributed nature. This validation is also needed to be carried out within the CONCERT framework, in order to guarantee that the information provided to visitors is as relevant as it can possibly be. This has direct implications for the future context model to choose (see Section 3.2.3).

- *Incompleteness and ambiguity*: Again, due to its distributed nature, contextual information is often incomplete and could also potentially be ambiguous. This is the case in sensor-based context awareness as well as in the model proposed in CONCERT. It may be that Web sources are not always available or if they are, they may not convey the correct information. In addition, the GPS sensor embedded within mobile devices may not be accurate.

- *Level of formality*: Users' context in general is a complex reality defined by real entities and the relations among them. To describe this as an abstraction in a model is a challenge, for many conceptions have to be shared and established beforehand. In ontology-based models, like the one suggested for CONCERT, the level of formality has direct implications on the kind of reasoning that can be performed.

- *Applicability to existing environments*: Ideally speaking any context model ought to be able to process any contextual information. As shown previously, the approaches that have been reviewed only run under very specific circumstances at particular locations. This is directly related to the two level interoperability idea presented at the beginning of this Chapter. On the one hand, interoperability at the model level is crucial in order to share and interchange information between context models, and on the other, interoperability at the infrastructure level will ease to use context-aware applications in any environment, in an attempt to make context-aware systems available in to the general public following the *anywhere* and *anytime* ideas that underline the ubiquitous computing paradigm.

- *Machine understandability*: Context information needs to be expressed in a way that is ready for machine consumption.

- *Accessibility*: Context information needs to be stored in sources that will be almost 100% available in order the context model to be as complete as possible and avoid incompleteness and ambiguity as much as possible.

- *Up to date*: Context information and tourism information will need to be continuously updated in order to produce the correct outputs and not discourage visitors that use a CONCERT framework based application.

The following table shows the implications of the requirements of context information in the subsequent context model:

Context Information Characteristics	**Context Modelling Requirements**
Distributed composition	The model needs to integrate various context information
Partial validation	The model needs to check its consistency
Incompleteness and ambiguity	The model needs to check its consistency
Level of formality	The model needs to check its consistency, reason and infer
Applicability to existing environments	The model needs to re-use, share and integrate context information
Machine Understandability	The model needs to process and "know" what it is processing
Accessibility	The model needs to avoid incompleteness and ambiguity
Up to date	The model needs to produce reliable results

Table 3.4: Implications of context information requirements in the context model

3.2.3 Requirements in terms of the context model

Models are needed in order to have some sort of representation about reality. For the case of the CONCERT framework, the context model is essential, for it represents the reality of visitors at a given moment of time in a way that mobile devices can compute the values of that reality and based on them, filter tourism incoming information.

One of the reasons, amongst others, for contextual computing not being more present in real life is because there are no standard context models or context management techniques[CK01] [Ay07] [BBH+09]. Most of the current approaches in recent research efforts focus on the development of uniform context models[Str03] [WZGP04] [GPZ94] [CFJ05], representation and query languages[SLPF03] as well as reasoning algorithms[CFJ04b] [WZGP04] that facilitate context sharing and interoperability of applications[SLP04] [CFJ04b] [GPZ94], i.e. interoperability at the model level.

Furthermore, contextual computing is gaining more importance in distributed systems and it is increasingly needed to exchange and integrate heterogeneous information stemming from distributed data sources. What is the model intended for? What is the model going to do with the context information? In which way is the model going to use context information?

Contextual information modelling is a key aspect not only for computing devices to efficiently access context, but also to achieve context information interoperability and an efficient information processing and managing. The most relevant context modelling approaches have been analyzed and summarized in different surveys[CK01][SLP04][BBH+09], whereby context models are classified by the scheme of data structures used for representing and exchanging contextual information in the respective system.

Summarizing, to choose one approach or another is not better or worse, it just determines the way the future system will work as well as its main functionalities. Therefore, one needs to really know the characteristics and functionalities of the application to build in order to identify its requirements and consequently, the context information model, or context management model.

The most important context models are summarized as shown in the following list[CK01][SLP04][Bal07][BBH+09]:

- *Key-value models*: These models represent the simplest data structure for context modelling. They are frequently used in various service frameworks where the key-value pairs are used to describe the capabilities of a service. This model can be used by providing the value of a context information to an application as an environment variable, e.g. location[SAW94].

- *Markup scheme models*: all markup based models use a hierarchical data structure consisting of markup tags with attributes and content. Typical representatives of this kind of context modelling approach are profiles. An example of this approach are the Comprehensive Structured Context Profiles (CSCP)[HBSS02].

- *Graphical models*: the Unified Modelling Language (UML) is also suitable for modelling context. Various approaches exist where contextual aspects are modelled in by using UML[BBH+09]. There are also other graphical approaches for context modelling based on

Context Modelling Language (CML)[HIR02]. CLM is based on Object-Role Modelling (ORM), which was developed for conceptual modelling of databases. CML provides a graphical notion designed to support software engineer analysing and formally specifying the context requirements of a context-aware application[BBH+09].

- *Spatial Models*: Space is regarded as an important context in most context-aware applications. In fact, most context definitions include the notion of space somehow, although the way it is used does not contribute to clarify the notion of context. Most spatial context models are fact-based models that organise their context information by physical location[BBH+09].

- *Object Oriented models*: to model context by the means of this abstraction offers the full power of object orientation, e.g. encapsulation, reusability, inheritance and various other typical aspects of object oriented (programming). Existing approaches use various objects to represent different context types and encapsulate the details of context processing and representation. Access the context and the context processing logic is provided by well-defined interfaces[BBH+09].

- *Logic-based Models*: logic based models have a high degree of formality. Typically, facts, expressions and rules are used to define a context model[DCGP+06][DC04][DC07].

- *Ontology-based models*: ontologies show a description of the concepts and their relationships. Therefore, ontologies are a very promising instrument for modelling contextual information due to their high and formal expressiveness and the possibilities for applying ontology reasoning techniques. Various context aware frameworks make use of ontologies as underlying context models, as shown in the previous Chapter.

Thus, due to (i) the definition of context that has been put forward earlier, (ii) the functionality foreseen for a CONCERT framework based application, (iii) the justification of the use of ontologies presented in Section 3.3 and table 3.4, (iv) the influences of the nature of context information on the model (shown in Section 3.2.2) and (v) the context information requirements reviewed in the previous section, ontologies seem to be also in this case the most adequate and appropriate assets to model context. Besides, the fact that context information will be retrieved from distributed heterogeneous Web sources emphasizes the requirement for interoperability and machine understandability of context information. Ontologies have proved to be good at this.

There is a need for a shared agreement on the notion of context and its constituents within the framework of human mobility in order to meet this requirement. Moreover, there is a need to have an intermediary infrastructure to harmonize data and make it machine understandable. Ontologies can act as data intermediaries and put together distributed heterogeneous information in a machine understandable way that allows different data sources to interoperate and exchange information for a consistent Context Model.

But there is more to the usage of ontologies than these reasons. For example, as well as humans (most of the times unconsciously) reason on the abstraction of reality they have formed in their minds, i.e. context, computing systems do need to process reasoning on their model in order to be able to automatically infer a higher degree of information.

Therefore, one key requirement to build contextual computing applications is to give computing systems the ability to understand the situational conditions (contextual information) of an individual in a way that it can undertake some kind of (useful) action for us humans, i.e. reasoning on that information. This implies that contextual information needs to be represented in ways that are adequate for this reasoning to occur, such as ontologies.

3.2.4 Requirements in terms of context information gathering

Another reason for context-aware applications not being more widely available is the lack of an adequate infrastructure[GPZ05] both in terms of hardware and software. It is somewhat surprising that no effort on research has been invested in investigation neither on new ways of gathering contextual information nor normalizing how contextual information ought to be obtained.

Most of the context-aware systems reviewed in Chapter 2 rely on networks of sensors to gather contextual information. Some of them use other gathering techniques to complement data provided by sensors, such as Web services. Sensing computing represents an obstacle on the one side to widely develop ubiquitous computing (in terms of preconditions for use and location of use) and higher complexity for developing context-aware systems, on the other.

One key assumption of CONCERT is not to use external sensors and to gather information from three different sources:

- *Internet*: General context information (e.g., weather conditions, environment, etc.) and tourism information. One key requirement of all of this information is that it needs to be accessible, retrievable and up to date, in order not to discourage visitors. Examples of such Websites could be the following:
  - Open Data Euskadi Initiative, http://opendata.euskadi.net/w79-home/eu.
  - Yahoo! Weather, http://developer.yahoo.com/weather/.
  - TripAdvisor, http://www.tripadvisor.com.
  - Pachube, http://www.pachube.com/.
- *Visitor*: Personal information, provided directly to mobile devices through the CONCERT application interface.
- *Mobile device embedded sensors*: These sensors can provide location data and in the case of advanced devices, orientation.

3.3 Justification of the Context ontology

At the beginning of this Chapter, the three main characteristics of the CONCERT framework have been stated, whereby ontologies will be used to model visitors' context. Once the notion of context within the framework of human mobility has been presented, it needs to be converted into a computing model in order to make it operative and provide context based information. In fact, one of the biggest advantages of ontologies is their flexibility and capability to determine and model a domain and, hence, conceptualize the portion of reality to which such a domain refers[TSSP08].

This Section presents the justification of the use of ontologies for context modelling from various perspectives, namely philosophical, functional, requirements and historic.

3.3.1 Philosophical justification

From a strictly theoretic point of view, based upon the original philosophical conception of Ontology according to Aristotle, the way in which science is communicated is based on language, i.e. semantics[Kli05]. So, Semantics in Aristotelian philosophy represent the relationship there is between the reality of things in the world (Aristotle's concept of Ontology) and the idea (model) that people form of them in their minds.

In this dissertation, a context model is a formal representation of an individual's context in a computing device. The model is specified using a set of concepts, properties and relations, i.e. ontologies. In addition to that, a context model is an abstraction of an individual's (situational) reality at a given moment of time. Thus, as well as humans use semantics (i.e. natural language) to explicitly express their idea about the world (relate their idea about the world with the world itself), ontology semantics (i.e. ontology representation languages) convey (the reality of) the model to a computing entity, i.e. the model and the computing entity are related through the context ontology.

This parallelism between the notion of Ontology in philosophy and the notion of ontology within computer science as a tool to model context, theoretically and conceptually grounds the use of ontologies to model context. The abstraction of reality (i.e. context of an individual at a given moment of time, Ontology) in a computing system can be represented through ontologies. The relationship that exists between the model (mental abstraction) and the reality is expressed through semantics (language)[Kli05] as well as the computing model of context (ontology) can be expressed through the (ontology's) semantics, i.e. ontology languages.

3.3.2 Functional justification

Ontologies have proven to be good intermediation tools in information integration. This is crucial within CONCERT's vision of context: this new vision of contextual computing has a new approach to the notion of context and sets two levels of interoperability, i.e. infrastructure and model interoperability, which is given by ontologies. Future semantic-based applications will be characterized for using a large number of ontologies that will be interconnected in networks of ontologies. Since, contextual computing will be one of these applications, the use of a network of ontologies will provide the model with the level of interoperability and scalability required.

Ontologies provide reasoning functionalities too. As well as humans (most of the times unconsciously) process reasoning activities on the abstraction of reality they have formed in their minds, i.e. on the information originated in their context, computing systems do need to process reasoning on their model in order to be able to automatically serve humans.

Therefore, one key requirement to build contextual computing applications is to give computing systems the ability to understand the situational conditions (contextual information) of an individual in a way that it can undertake some kind of (useful) action for us humans, i.e. reasoning on that information. This implies that contextual information needs to be represented in ways that are adequate for this reasoning to occur, such as ontologies.

In addition, ontologies are very powerful tools to specify concepts and interrelationships among them. The real power of ontologies lies in fact in the possibility to define and build relationships between terms and properties of the terms that are part of the ontology and in the reasoning capability derived from these relationships. They provide the means to model real-life situations into machine understandable data structures and allow humans and machines to work in cooperation[CFJ04b]. For context is also a system of concepts and relations, ontologies arise as a possible means for context modelling. Contextual knowledge can be interpreted and evaluated by the use of ontologies. This allows computers to determine contextual compatibility to compare contextual facts and to infer new and more complex context from core measurements. Consequently, the use of ontologies to model and manage contextual information entails various advantages with respect to other existing approaches.

After having analyzed ontologies' main applications according to[UG02] three main types of uses can be identified for ontologies:

- *Knowledge communication and sharing*: ontologies may be used as a communication vocabulary for various agents, either machines or humans.

- *Logic inference and reasoning*: ontologies may be used to infer implicit knowledge based upon explicit knowledge and some rules or logic inferences procedures. These kinds of applications permit the creation of ubiquitous ambient intelligence systems.

- *Knowledge re-use*: the use of ontologies and their hierarchical relations allow to re-use general ontologies in the definition of ontologies of particular domains. This entails a substantial advance with respect to other modelling techniques, since it allows the definition of standard ontologies used in intelligent systems.

3.3.3 Requirements justification

At the end of their survey Strang and colleagues[SLP04] concluded that ontologies are the most suitable approach to model context according to the requirements they had previously set. Furthermore, ontologies have been widely accepted as instruments to model context[KS07].

The requirements that have been identified for context information can be handled with ontologies in the following way:

- *Distributed Composition*: The CONCERT contextual computing framework will use distributed and heterogeneous information as stated previously. As well as it happens with sensor based context-aware applications, in spite of CONCERT gathering context information from the Internet, this information is in its nature distributed and heterogeneous. Ontologies have proved to be comprehensive data mediation and integration tools that allow interoperability among systems. This would very much benefit to access service information in different information sources through the context ontology.

- *Partial Validation*: In CONCERT, information will have to be validated as well by the context model. Ontologies can be used for this purpose and due to their reasoning capabilities.

- *Richness and quality of information*: Strang and colleagues[SLP04] assume that contextual computing systems under study in their survey gather contextual information from a network of interconnected sensors. These sensors deliver different quality of data over time and this can be a problem. This problem shall be used by not using sensors, as it has been mentioned earlier in this chapter: our system will gather contextual information from already existing Web-based (distributed and heterogeneous) information sources. Ontologies are a crucial tool at this point in order to meaningfully make use of contextual information. We shall have this challenge as well. In our case, instead of actually communicating data coming from a particular sensor, we will have to rely on the administrator of a particular Website. The information provided by this site needs to periodically be updated in order to provide real and true information. The mobile device incorporated GPS sensor will also have to be well calibrated.

- *Incompleteness and ambiguity*: Due to the fact that context ontologies have explicit representation of their semantics (due to the nature itself of ontologies) they can be reasoned by the available logic inference engines. This means that implicit context information can be inferred from explicit information. Ontology's reasoning capabilities will allow to (automatically) find context information incompleteness and inconsistencies and lack of robustness. Traceability of information is also very important, especially with regard to data inferred to form high-order context. In this case, it is fundamental to know the derivation (inference) rules used. The robustness of the model will partially be determined by its ability to overcome this problem, e.g. by detecting potential data inconsistencies. Data inconsistencies can also occur if information is coming from Web sources. Ontologies are a good means to detect these kinds of problems due to their reasoning capabilities. They provide a high degree of robustness to the whole system in terms of reasoning capabilities.

- *History, logging*: It is out of the scope of this dissertation, however history of context information is something very important as some decisions sometimes depend on past events. We will establish the way to store contextual information, although we will not focus on how nor what that information may be used for.

- *Level of formality*: Again, ontologies are an appropriate approach to model context not only for technical and practical reasons based upon ontologies' functionalities, but also for theoretical grounded reasons. Ontology description languages also provide the means to define things (contexts) with different levels of formality that imply different levels of expressivity of the model. We have to reach a minimum level of agreement within a sufficiently representative number of members within the community in order to establish a particular ontology language for ontology development. This way, we guarantee that all ontologies (expressed with RDF[W3C04c] and OWL[W3C04b]) will have the same level of formality and expressivity and will be comparable, re-usable across applications and will be able to work in co-operation. Ontologies expressed in these languages provide the means for independently developed systems to share context knowledge. In fact, ontology definition languages have been supported and approved as de facto standards in the computer science domain by the W3C. This way, ontologies may be used in any kind of machine.

- *Applicability to existing environments*: ontologies support universality in terms of being re-usable in different systems provided they had been defined with the same degree of formality. Furthermore, our conception of context puts the human in the centre of the system. Our objective is not to model a particular location or a particular meeting room as can be read in the literature. By contextualizing the particular individual at a particular location by the use of Web-based existent information we are giving one step forward to make contextual computing systems and applications universal and applicable in already existing environments.

This is one of the most challenging aspects of ubiquitous computing and contextual computing applications and the reason why:

- This new approach proposes to contextualize the traveller instead of contextualizing a particular environment.
- This new approach intends to use Web information to model the traveller. The Web is accessible almost anywhere through connectivity technologies, whereas sensors are not.
- This new approach suggests to use ontologies to model context: they are the means to integrate the necessary information needed in context models anywhere, for ontologies are proved intermediation tools[FHKL09].

3.3.4 Historic justification

The decision of using ontologies for the CONCERT context model is also supported by the experience of other authors, as shown in Chapter 2. For example, Chen and colleagues[CF03][CFJ04b][CFJ05] argue that the use of ontologies is a key requirement for the realization of the pervasive computing vision. They make intensive use of ontologies in order to address context modelling and reasoning as well as knowledge sharing and user privacy will enhance their context-based intelligent software agent application's functionality. The use of context ontologies is also proposed for context sharing and interoperability in ubiquitous and pervasive computing environments and systems[CFJ04b].

Moreover, ontologies may well support independently developed sensors, devices and agents share contextual knowledge and to provide relevant services and information to users based on their situational needs[Che03].

On top of these general benefits of using ontologies, the following list summarizes further reasons to use them for contextual computing applications[Che03]:

- Ontologies give meaning to information besides the contextual computing system that utilizes that information. This allows contextual data sharing and interoperability across systems.
- Languages used to define ontologies provide high expressivity to information. This provides the means to adequately model different kinds of (contextual) data in terms of their nature: time, people, locations, etc.

- Ontologies provide inference engines that by the use of logic rules obtain further (inferred) contextual information. This is extremely valid to detect data inconsistencies as well as to infer complex contextual information that may firstly be implicit.
- Besides, ontologies may be used to express, store and interchange other kind of information augmenting the interoperability of a ubiquitous environment or systems through unified data formats.

Gu and colleagues argue that, given the lack of common and adequate infrastructure to support context-aware systems[GPZ94][CK01], their architecture needs to support a common context model that can be shared by all devices and services (by the use of ontologies, due to their nature) and a set of services that perform context acquisition, context discovery, context interpretation and context dissemination.

McGrath and colleagues[MRCM03][MCM03] argue that ontologies are valid for Contextual Computing applications because they provide the means to:

- Validate during runtime consistency between description of different entities and axioms defined in the ontologies. This implies that operative parameters with regard to security and environment are fulfilled.
- Implement semantic discovery mechanisms of entities. Ontologies support the discovery of ubiquitous resources by means of semantic descriptors independently from their identifiers or network addresses.
- Facilitate users to better understand the environment's logic structuring and how entities within the environment relate among them and with the user.
- Both users and software agents undertake search process in the system and easily interact with the various components of the system.
- Create behaviour rules that modify the environment on basis of context acquired by it. These rules may be created both by humans or software agents.
- Add new components to the system in an easy manner. The definition of ontologies oriented towards interoperability allows different applications or ubiquitous systems to interact among them.

3.4 Summary

According to the literature review, the notion of context in the framework of human mobility has been directly imported from human computer interaction without specifically addressing its characteristic requirements. This lack of a sufficient theoretic support has lead to an oversimplified understanding of context. This Section presents the summary and conclusions extracted from the theoretic contributions provided by this Chapter.

The study of the notion of the nature of context in the framework of human mobility presented at the beginning of the Chapter has lead to provide new definitions. They are not intended to be the *final* definitions of context, but rather they intend to be an operative and integrative definition of context valid for human mobility. In this sense, new definitions for the notions of *contextual computing in tourism* and *context* have been provided.

Aiming at reaching a higher degree of interoperability, the concept of *two level of interoperability* has been introduced, i.e. it is not only sufficient to provide interoperability at a model level, but also at the infrastructure level, avoiding this way hardware sensors and making this way contextual computing applications in tourism independent from location while maintaining this variable in the model. This has major implications in terms of context information gathering, whereas interestingly, context modelling requirements are not so much influenced by the type (hard vs. soft) of context information sources.

All the above has direct implications in the context modelling requirements in terms of type of information, nature of that information, type of context model and the way context information is going to be gathered. Some of the existing requirements for previous context applications are still valid for this one, but in addition, new characteristic requirements typical from human mobility have been added.

The need to exchange context information, make it machine understandable and processable, interoperability and reasoning (for data reasoning and consistency check) pose also tight requirements in terms of context modelling. Using ontologies to model context has proved to be appropriate to achieve these goals. A Section of this Chapter is devoted to justify the use of a network of ontologies in this regard from various perspectives: philosophic, practical, and history-based.

This network of ontologies incorporates the requirements of the framework of human mobility and implements them in terms of motivation, preferences-demographics and role according to standard parameters established by the tourism scientific community[UNW08b] as will be shown in Chapter 4. A rule-based information engine built on top of the ontology will filter the incoming tourism information with respect to context values at a given moment and will select the one that best matches with visitors' context to support their mobility. These two issues together with the definition of the notion of context that is suggested later in this Chapter will be crucial to define CONCERT's future functionality (Chapter 5)[Str03].

Chapter 4

ContOlogy: The Context Network of Ontologies

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."
Edsger Wybe Dijkstra, (1930 - 2002), Dutch Computer Scientist.

As Section 3.5 indicates, it is necessary to use ontologies to model the notion of context put forward in the previous Chapter. This will allow to not only achieve CONCERT's full functionality, but also to enhance interoperability of context information at the model level (Chapter 3).

Chapter 2 has shown a number of context modelling ontologies[CFJ05] [GPZ94] [WZGP04] [PVR+06] [SRA06]. However, these ontologies are still in an early experimental phase and in addition, their use is very dependent on the particularities related to their application domain and techniques used. Thus, despite the fact that information can be integrated in those context models (due to the use of ontologies), the ontologies themselves are hardly implementable in other applications with distinct objectives. Moreover, their extension is not obvious either without impacting on the model's functionality. Furthermore, since supporting humans on the move at a large scale (as intended in CONCERT) was not within the objectives of the projects in which these ontologies were developed, therefore, their (straight forward) use in CONCERT would not guarantee addressing all the context information requirements specified in Section 3.2.

Therefore, a new ontological approach is needed to tackle these issues. The ontology development in the CONCERT framework is a critical step, since not only does interoperability at the model level have to be guaranteed, but also the whole model itself needs to be as flexible as possible and needs to cover the typical information requirements of a human mobility environment, as specified in the previous chapter. One possible approach to overcome these obstacles is to use networks of ontologies[HRW+06], whereby a network of ontologies is a collection of ontologies that are related to one another by properties. This way, the context model proposed in this dissertation would not just provide interoperability at the model level (for using ontologies) but in addition, the use of networks of ontologies would enhance the model's modularity and flexibility, hence its interoperability and re-use would be much simpler and less dependent on the specific purpose for which the ontology was built[SFFLGP+08]. In addition, the specific information requirements related to human mobility will be carefully addressed in order to assure that the network of ontologies is actually valid for CONCERT's purposes.

Choosing the right methodology to build ontologies is crucial. Thus, the first Section of the Chapter makes a quick analysis of the existing methodologies in Section 4.1. A brief conclusion of this review shows that whichever the methodology chosen, it has to address the particularities of the nature of the future network of ontologies and the way it is intended to be built, i.e. re-using and reengineering already existing ontological resources, for example. The only known methodology that considers these (and other potentially interesting) issues is the NeOn methodology for building networks of ontologies[SFFLGP+08]. This methodology and its main characteristics will be described in Section 4.2.

The following section, Section 4.3, provides some useful definitions and terminology established by the IEEE related to software development. Then, Section 4.4, thoroughly justifies the use of the NeOn methodology for building CONCERT's network of ontologies.

Based on the previous, Section 4.5 thoroughly describes step by step the development of the ContOlogy network of ontologies[LASM+09b]. The outcome of this process is presented in Section 4.6, where the ContOlogy network of ontologies is described.

Section 4.7 is devoted to the evaluation of the network of ontologies. And finally, Section 4.8 will conclude this chapter with a brief summary.

4.1 Ontology building methodologies

Researchers, academics and practitioners in general have especially been interested in approaches for building ontologies (sometimes from scratch), for re-using other ontologies and for using semiautomatic methods that reduce the knowledge acquisition bottleneck of the ontology development process[GPFLC03]. In fact, during the first years of intensive use of ontologies, their development was rather random and had neither theoretic support nor guidelines.

Thus, when ontologies started to be widely used for multiple applications in a number of different domains in the early 1990s, there was no common methodology to support the ontology development process. Each research team followed their own criteria to design and later manually build the ontologies. However, some general steps and interesting points about the Cyc ontology development process were published in 1990[LG90]. Since then, authors that have conducted research on ontologies and needed to build one also published the steps they followed to create the ontology as a new ontology building methodology[LG90][SFdCB+08].

In this same way, Gruber argued that formal ontologies need to be designed and therefore, the criteria needs to be grounded on the purpose of the resulting final ontology. Thus, he enumerated a set of design criteria (based just on 5 steps) to be followed by researchers that wanted to build ontologies for knowledge sharing and interoperation among programmes based on shared conceptualizations[Gru94] as follows:

1. *Clarity*: ontologies need to effectively communicate the intended meaning of defined terms.

2. *Coherence*: it should sanction inferences that are consistent with the definition.

3. *Extendibility*: ontologies should be designed to anticipate uses of the shared vocabulary.

4. *Minimal encoding bias*: the conceptualization of the ontology should be specified at the knowledge level[New80] without depending on a particular symbol-level encoding.

5. *Minimal ontological commitment*: sufficient to support the intended knowledge sharing activities.

Grüninger proposed other guidelines to build ontologies in 1995 in the framework of the TOVE (Toronto Virtual Enterprise) Enterprise Modelling project[GF95] based on scenario motivations that arise in the applications. These have the form of story problems or examples that are not specifically addressed by existing ontologies. The motivating scenario can also provide a set of intuitive solutions for the problems.

The following year, Uschold and Grüninger proposed another methodology for building ontologies[Usc96b] arguing that despite the fact that there was (at that time) a significant amount of work going on developing ontologies, there was not an agreed method for building them. They proposed a method based on identifying the level of formality, scope and the purpose of the ontology, i.e. why the ontology is being built and what its intended uses are. Once these two aspects had been made clear, the ontology building processes would be based on ontology capture (identifying the key concepts and relationships in the domain of interest), ontology coding and integration with existing ontologies.

Some years afterwards, in 2001, Natalya F. Noy and her colleague Deborah MacGuinness proposed a 7 step based methodology to create an ontology[NM01]:

1. *Step 1*: As well as Grüninger, they proposed to first identify the domain and the scope of the ontology using the competency questions[GF95], i.e. a knowledge-base based on the ontology ought to be able to answer.

2. *Step 2*: Re-use existing ontologies.

3. *Step 3*: Enumeration of the important terms of the ontology.

4. *Step 4*: Definition of the classes and their hierarchies.

5. *Step 5*: Definition of the slots and relationships of the classes.

6. *Step 6*: Definition of facets of slots.

7. *Step 7*: Population of the ontology with instances, i.e. particular objects.

Bernaras and her colleagues published yet another method for building ontologies in the domain of electrical networks[BLC96]. They recommended several steps to follow within the construction phase:

1. *Specification of the application*: this phase consists of providing an application context and a view of the potential components of the ontology depending on its application domain.

2. *Preliminary design*: based on relevant top-level ontological categories. The input for this phase is formed by a list of terms and tasks developed during the previous phase.

3. *Ontology refinement and structuring*: based on modularisation and ontology hierarchy organisation.

So far, the previous methodologies are valid for building ontologies from scratch. However, the number of ontologies started to increase dramatically at the end of the 90s and re-using part of them seemed a good idea. This way, the METHONTOLOGY methodology for building ontologies was presented as well in 1991[GPFLC03]. The METHONTOLOGY methodology presents some guidelines to build ontologies either from scratch, by re-using other (already existing) ontologies or by a process of ontological reengineering[CFLGP01], which marks a great difference with existing methodologies. METHONTOLOGY enables the construction of ontologies at the knowledge level[CFLGP01][GPFLC03][SFFLGP$^+$08]. METHONTOLOGY was also "revolutionary" at that time in that it tries to apply typical software development guidelines in ontology building, thus it identifies a set of activities to be carried out based on the main activities identified by the software development process and used in Knowledge Engineering[GPFLP97][Wat86]. This includes: the identification of the ontology development process, a life cycle based upon evolving prototypes, and techniques to carry out each activity in the management, development-oriented and support activities.

Corcho and colleagues proposed a new methodology called On-To-Knowledge[CFLGP01]. The objective of both the On-To-Knowledge project and the methodology designed within the framework of the project was to apply ontologies to electronically available information in order to improve the quality of knowledge management in large and distributed organizations[CFLGP01]. Tools and methods were developed to access large volumes of semi-structured and textual information sources in intra-, extra- and internet-based environments.

The methodology includes guidelines for building ontologies to be used by the knowledge management application and introducing knowledge management tools into companies, supporting and assisting knowledge providers and seekers to present knowledge in an efficient and effective manner. Therefore, the On-To-Knowledge methodology for building ontologies proposes to build ontologies taking into account how they are going to be used in further applications, i.e. the methodology is based on the analysis of usage scenarios (scenario driven)[SFFLGP$^+$08].

4.2 NeOn methodology for building Networks of ontologies

Section 3.3 is devoted to thoroughly provide arguments and facts to use ontologies in the framework of CONCERT. Furthermore, the Section also specifies all context information requirements and the context model's main characteristics. These, as stated earlier, have a direct influence on the context ontology to be implemented. Based on this, as mentioned in the introduction of this Chapter, a new ontological approach is going to be implemented for the context model in the CONCERT framework, by means of a network of ontologies. Important characteristics of networks of ontologies include versioning, importing other ontologies, extending other ontologies, modules of other ontologies and mappings with other ontologies.

The methods and methodologies presented in Section 4.1 do not support the development process of an ontology with the functionalities and characteristics required for the network of ontologies of the CONCERT framework and furthermore, they do not consider collaborative and distributed development of ontologies[SFdCB+08] [GPFLC03]. In fact, the first methodology that included a proposal for collaborative construction of ontologies was the so called CO4 methodology[Euz95] [Euz96] [CVGL09] [CRL+09]. Some years afterwards, the DILIGENT methodology was proposed to support domain experts in a distributed environment to engineer and evolve ontologies[PTS04]. This methodology is focused on collaborative ontology engineering and the central issue is to keep track of the change arguments. The ontology building process put forward in the methodology suggests the following 5 main activities: Build, Local adaptation, Analysis, Revision and Local Update[PTS04].

The NeON methodology for building ontologies (networks of ontologies) [SFdCB+08] covers the gaps and limitations of the previous methodologies, i.e. lack of guidelines for building ontologies by re-using and reengineering other ontologies and knowledge, lack of guidelines for contextualizing an existing ontology and plugging it in with existing ontologies that might be in continuous evolution. Finally, the existing methodologies do not explain the ontology development process with the same granularity as software developing methodologies.

The ontology presented in this dissertation has not been built collaboratively, however, the rest of the features of the NeOn methodology are not only interesting from a methodological point of view, but necessary to build the ontology that is presented in this dissertation: the ContOlogy context network of ontologies will be developed both by re-using already existing ontological resources that adequately fulfil the requirements specified in the previous Chapter (Section 3.2) and by building ontologies from scratch. The whole process of building the network of ontologies shall be treated as a software development process, which scope is set in Section 4.3 by providing the definition of certain terms.

The following three sections will describe more in detail the NeOn methodology for building networks of ontologies.

4.2.1 Scenarios for building ontologies

Experience and revision of the different ontology building methodologies show that there are different ways to develop them. These ways can be seen as different scenarios (see figure 4.1) for which the NeOn methodology accounts the following[GPSF09]:

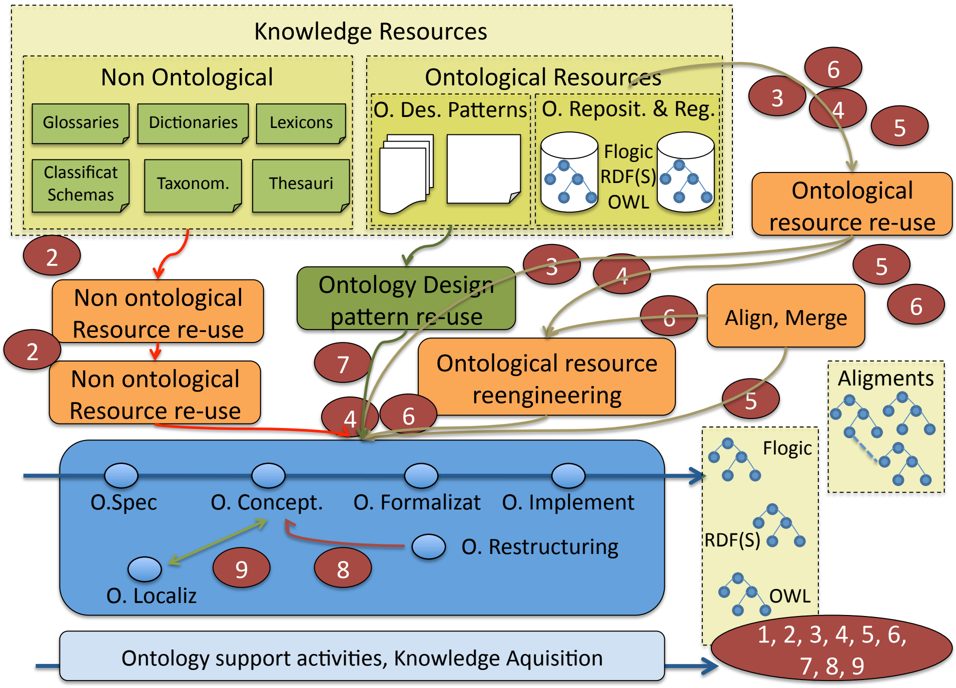

Figure 4.1: The NeOn scenarios for building networks of ontologies

- *Scenario 1*: Building networks of ontologies from scratch without re-using existing knowledge resources.

- *Scenario 2*: Building networks of ontologies by re-using and reengineering non ontological resources.
- *Scenario 3*: Building networks of ontologies by re-using ontological resources.
- *Scenario 4*: Building networks of ontologies by re-using and reengineering ontological resources.
- *Scenario 5*: Building networks of ontologies by re-using and merging ontological resources.
- *Scenario 6*: Building networks of ontologies by re-using, merging and reengineering ontological resources.
- *Scenario 7*: Building networks of ontologies by re-using ontology design patterns.
- *Scenario 8*: Building networks of ontologies by restructuring ontological resources.
- *Scenario 9*: Building networks of ontologies by localizing ontological resources.

The ontology to be built does not necessarily have to follow one or another of the scenarios defined above. On the contrary, the ontology building process is a combination of a number of scenarios. Scenario 1 is one of the most typical, i.e. building ontologies from scratch. In fact, all of the methodologies that have been referenced before in Section 4.1 have been designed for this purpose.

4.2.2 Ontology Specification

The ontology specification refers to a collection of requirements the ontology ought to fulfil. These requirements are all gathered in a document called Ontology Requirements Specification Document (ORSD) that includes the purpose, level of formality and scope of the ontology, intended users and uses.

Various different techniques and ways have been published in the literature to elucidate the specification of ontologies. Grüninger and Fox[GF95], Noy and MacGuinness[NM01] and other authors propose the following steps to formally specify ontologies:

- Identify the purpose the ontology is going to be used for.

- Identify the intended uses of the ontology.
- Identify the requirements the ontology needs to fulfil after having been formally implemented.

There are different techniques to accomplish this task, i.e. identification of requirements. Some of the most commonly used are brainstorming sessions between ontology developers, users and software engineers. Other techniques use the exploitation of the use cases in which the ontology is going to be used. However, most of the existing methodologies suggest using the so called competency questions[GF95]. Competency questions (CQs) are natural language-based questions that the ontology needs to answer once it has been developed.

Based on the existing methodologies, the NeOn methodology proposes the following guidelines to find out the requirements of the network of ontologies[SFFLGP+08]:

1. *Task 1*: Identify purpose, scope and level of formality.
2. *Task 2*: Identify intended users.
3. *Task 3*: Identify intended uses.
4. *Task 4*: Identify requirements.
5. *Task 5*: Group requirements.
6. *Task 6*: Validate requirements.
7. *Task 7*: Prioritize requirements.
8. *Task 8*: Extract terminology and frequency.

4.2.3 Scheduling and identifying the ontology life-cycle

Once the specification of requirements has been obtained complying the previous tasks, the rest of the activities within the development process have to be identified and scheduled[SFFLGP+08][SFGP08].

This is accomplished in the NeOn methodology as follows:

- Find out the life cycle model of the network of ontologies.
- Select processes and activities.

Author	**Name of method.**	**Approach**	**Comments**
Gruber	N/A	5 step approach	from scratch building
Grüninger	N/A	Level of formality, scope	Use of CQs
Noy et al.	N/A	7 step approach	from scratch building
Fdz. López et al.	METHONTOLOGY	ontologies as SW artefacts, knowledge level	Scratch, re-use, reengineering
Corcho et al.	On-To-Knowledge	scenario exploitation	
Pinto et al.	Diligent	5 step	Collaborative, track of changes
Suárez Fig. et al.	NeOn	Scenarios, reqs., ontologies as SW artefacts	Networks of ontologies, versioning, scalability, systematic

Table 4.1: Ontology building methodologies

- Map processes and activities onto the life cycle model to obtain the life cycle of the network of ontologies.
- Establish the sequence and priority of the activities and processes.
- Establish the constraints in resources.

This whole process is thoroughly described and specified in Section 4.5.

Table 4.1 shows an overview of the revised ontology building methodologies.

4.3 Definitions

Methodological frameworks are widely accepted in different mature fields, such as Software Engineering. Taking into account that single ontologies and networks of ontologies could be seen as software entities[SFdCB+08], current software developing methodologies could be relevant to and applied within ontology engineering. Therefore, considering this, the NeOn Methodology sticks to the Institute of Electrical and Electronics Engineers[1] (IEEE) standard definitions related to software development processes, life cycle models and life cycle in Software Engineering.

Since the development of the ContOlogy network of ontologies shall follow the guidelines proposed by the NeOn methodology, it will also stick to these same IEEE definitions in order to avoid any possible misunderstanding. Therefore, the building of the ContOlogy context modelling and management network of ontologies shall be considered as a specific case of development of a particular software entity.

The terms methodology, method, technique, process, activity and so forth, have often been used in a vague and not very rigorous manner and thus, their meaning varies depending on the authors, papers and/or research projects. The following are the definitions provided by the IEEE.

The IEEE defines methodology as[IEE90]:

"A comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought-intensive work ought to be performed".

Methods and techniques are parts of methodologies. Thus, a method is defined as[BFLGPGP98]:

"A set of orderly processes or procedures used in the engineering of a product or performing a service".

A technique is[IEE90]:

"A technical and managerial procedure used to achieve a given objective".

Methods and techniques are strongly related because both are used to carry out tasks inside the different processes of which a methodology consists of. The IEEE defines a process as a [IEE96]:

"Function that must be performed in the SW life cycle. A process is composed of activities".

An Activity is a[IEE96]:

"Defined body of work that is to be performed, including its required input and output information."

[1] http://www.ieee.org

A task is the smallest unit of work subject to management accountability and according to the IEEE is defined as follows[IEE96]:

"A task is a well defined work assignment for one or more project members. Related tasks are usually grouped to form activities".

The relationships between the aforementioned definitions are summarized in figure 4.2. It can be seen that a methodology is composed of methods and techniques. Methods are further composed of processes on the one hand, and detailed with techniques, on the other. Processes can be decomposed in activities and finally, activities are made up of groups of tasks.

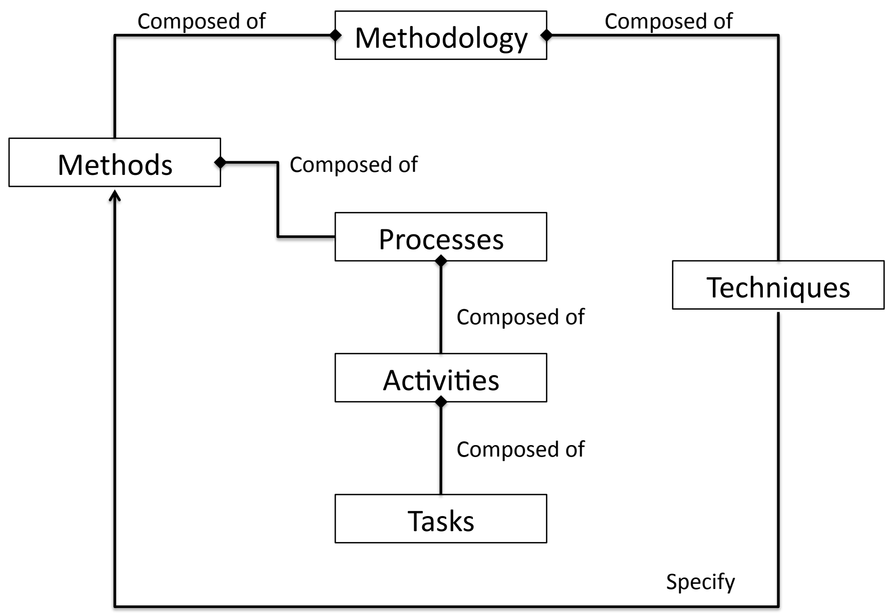

Figure 4.2: Relations between processes, tasks

4.4 Justification of the methodological support

Previous sections in this dissertation have justified the utilization of ontologies to manage context. The issue now is not only to decide how that ontology is going to be developed or under which methodology criteria, but to consider the constituents of that ontology in terms of classes, properties, relations and restrictions as well as the morphology of that ontology, i.e. whether it is going to be one single ontology or there are other possible associations that are more suitable to this particular case and to tourism in general.

As justified in Chapter 3, the single fact of pursuing context information interoperability and re-usability at the model level, makes it necessary to use ontologies. However, as demonstrated in Chapter 2 and shown by evidence of experience, other existent context modelling approaches based on the intensive use of ontologies have failed to be interoperable, since context-based systems do not exist for the general public. One of the possible reasons for this is that the particularities and structures of the ontologies that were proposed are highly dependant on the infrastructure, characteristics, application domain and techniques used to develop them.

The context modelling ontology proposed in this dissertation has to be interoperable with other ontologies wherever visitors may be. This means that there is a part of the context that is taken by and with visitors wherever they travel (personal sphere) and there is another part of context that has to be linked to the first, like for example the tourism services offered in a certain environment. This link or alignment can be improved or made easier with networks of ontologies, since versioning, formality and expressivity issues are considered when connecting ontologies within a network.

All of the previously mentioned statements together with the fact that a network of ontologies is planed to be developed has lead to consider looking for alternative methodologies, other to those presented in Section 4.1. The NeOn Methodology for developing networks of ontologies clearly fulfils the requirements for the following reasons:

- Ontologies in general have been so widely used that they have started to be considered software artefacts. However, no institution or body has normalized their development and maintenance and therefore, software engineers have applied their own criteria for these purposes. There is a real need in ontological engineering for methodological guidelines that support the whole life cycle of ontologies[SFdCB+08]. The NeOn methodology sticks to general IEEE software development standards and provides a normalized and systematized framework for developing networks of ontologies[SFFLGP+08].

- The need for an ontological structure that supports the modular conception of context provided earlier. The idea that underlies the notion of network of ontologies matches well with the (modular) vision of the notion of context presented in Chapter 3. Each of the constituents of context can be characterized by one ontology and following, they can be linked with each other forming a network of ontologies.

- NeOn provides guidelines to re-use and reengineer already existing ontological resources. This is important in itself, as the intention in this dissertation is not to re-do already existing work neither in ontology development nor in context management ontologies. Previous Chapters have described some of the existing context ontologies and other relevant ontologies to model and manage context. They can all be re-used in a way that is useful for the context model presented in this dissertation. However, in order to do so, guidelines and recommendations are required in order to do it in a coherent and consistent way.

The ContOlogy network of ontologies (at this first step at least) is not going to be built in a collaborative way. However, given that the notion of context provided earlier is plural and adapted to human mobility, scalable and very modular, it may be likely that in the near future somebody, somewhere may want to alter one (or more) of the original ontologies or may even want to add a new particular one on to the network. Therefore, the use of NeOn will guarantee this be done correctly in the future.

4.5 ContOlogy: Description of the building process

As stated earlier, one of the most recent methodologies and possibly the only one that provides framework support and guidelines to develop networks of ontologies is the NeOn Methodology[SFdCB+08]. Section 4.2 has presented and outlined the NeOn methodology in a general way. That same methodology is now specified, step by step, for the particular case of the ContOlogy context modelling and management network of ontologies building process.

The NeOn methodology for building networks of ontologies has been adapted to the particular needs and requirements of the CONCERT framework for contextual computing in tourism.

The sequence of tasks and processes chosen in this case have been as shown:

1. Selection of the scenarios for building the ContOlogy network of ontologies.
2. Specification of the ContOlogy network of ontologies.
3. Selection of processes and activities (based on the scenarios).
4. Planning and selection of the ContOlogy network of ontologies' life cycle.
5. Map processes and activities into the network of ontologies' life cycle.
6. Set the sequence of processes and activities.
7. Determine the resources and time constraints.

4.5.1 Scenarios for building the ContOlogy network of ontologies

The main objective of this task is to select the set of processes and activities that need to be carried out in the ontology building process. In order to do so, the NeOn methodology proposes a natural language-based questionnaire that needs to be answered in advanced and that will allow to choose the scenarios for building networks of ontologies.

There is one question per scenario except for scenario 1. The implementation of this scenario is compulsory in all developments of networks of ontologies, since it contains the fundamental activities such as requirements' specification, conceptualization, implementation and others.

The NeOn questionnaire and the corresponding answers to each question within the development of the ContOlogy network of ontologies are the following:

- Are any non-ontological resources planned to be used such as thesauri, data bases, etc., in the development of this network of ontologies?
  - If yes, then the development of the network of ontologies should follow scenario 2.
  - Answer: No. Only ontological resources are planed to be used.
- Are any existing ontological resources planned to be used in the development of this network of ontologies?
  - If yes, then the development of the network of ontologies should follow **scenario 3**.

  - Answer: Yes. The literature review detailed in Chapter 2 shows that there are already some authors who have attempted to model context by the means of ontologies. Some of these (already existing) ontologies may be of interest for the ContOlogy network of ontologies. The ContOlogy network of ontologies plans not only to use ontologies or part of ontologies that are created specifically to fulfil the requirements of the system presented in this dissertation, but also to re-use already existing ontologies in the realm of context-awareness.

- Are any existing ontological resources planned to be used and modified in the development of this network of ontologies?
  - If yes, then the development of the network of ontologies should follow **scenario 4**.
  - Answer: Yes. It may be that some of the elements of the existing ontologies may not be of interest, or the ones that are of interest may need to be modified in order to better adequate them to the requirements of the ContOlogy network of ontologies.

- Is any set of existing ontological resources planned to be used and/or merged in the development of this network of ontologies?
  - If yes, then the development of the network of ontologies should follow scenario 5.
  - Answer: No. The ContOlogy network of ontologies will only use or modify existing ontologies, it will not merge with any other existing ontology.

- Is any set of existing ontological resources planned to be used, merged and modified in the development of this network of ontologies?
  - If yes, then the development of this network of ontologies should follow scenario 6
  - Answer: No. Only ontological resources are planned to be used and modified in the development of the ContOlogy network of ontologies (scenario 4).

- Is any ontological design pattern planned to be used in the development of this network of ontologies?
  - If yes, then the development of this network of ontologies should follow **scenario 7**.

- Answer: Yes. The re-use of ontological design patterns refers to using available ontology design patterns in the solution of different modelling problems during the development of new ontologies[SFFLGP+08].

- Is restructuring of this network of ontologies planned?
  - If yes, then the development of this network of ontologies should follow **scenario 8**.
  - Answer: Yes. The structure of the final ontology will be adapted to the particular needs and requirements of this network of ontologies.

- Is the development of this network of ontologies planned to be in different natural languages?
  - If yes, then the development of this network of ontologies should follow scenario 9.
  - Answer: No. The network of ontologies will be fully developed in English.

Thus, according to (i) the answers provided to the previous questions and (ii) considering that Scenario 1, which is basic in the development of networks of ontologies has to be combined with the rest of the Scenarios, the following scenarios will be followed to build the ContOlogy network of ontologies:

- *Scenario 1*: development of networks of ontologies from specification to implementation.
- *Scenario 3*: Development of networks of ontologies re-using ontological resources.
- *Scenario 4*: Development of networks of ontologies re-using and reengineering ontological resources.
- *Scenario 7*: Development of networks of ontologies re-using ontological design patterns.
- *Scenario 8*: Development of network of ontologies restructuring ontological resources.

4.5.2 Specification of ContOlogy

Once the scenarios have been already identified, the next step within the process of building the network of ontologies is to specify the ontology, i.e. the ontology specification. The goal of this task is to identify the collection of requirements the ontology needs to fulfil[SFdCB+07]. The output of this activity is the ontology requirements specification document (ORDS), shown in Appendix I, that includes important aspects related to the ontology such as purpose, level of formality, scope of the ontology, target group and intended uses and finally, set of requirements, which are those that the network of ontologies needs to necessarily cover[SFdCB+08].

The following sections describe the tasks as specified in the NeOn methodology and the actions carried out in order to elucidate the collection of requirements of the ContOlogy network of ontologies.

Task 1: Purpose, scope and level of formality of ContOlogy

The development of ContOlogy is motivated by the need to build a new context model based on the requirements and theoretic contributions presented in Chapter 3. The lack of consensus both on the definition of the notion of context and on establishing a context modelling and management method have originated a number of different (non sufficiently established) context models and definitions.

Therefore, in order to address these limitations, a new framework for contextual computing in the realm of tourism has been suggested. In practical terms, this has implied a new definition of context, as well as a new set of context information modelling and management requirements, (presented in the previous chapter).

Therefore, the purpose of building the ContOlogy network of ontologies is to provide:

- An adequate framework for context management in the realm of tourism.
- Consensus on the way context is modelled and managed in the realm of tourism.
- An adequate framework to provide contextual based services in tourism.
- A modular conception of the notion of context that derives in:
  - A set of modular and scalable ontologies that are defined in the OWL DL standard language[W3C04b].

  - An easily extensible network of ontologies that represent other domains of interest within the travel and tourism domain that are also expressed in OWL DL.

- A step forward to make contextual computing services in tourism available to the general public in an efficient and sustainable manner.

Task 2: Identify intended users

The goal of this task is to establish who will the main intended users of the ontology be. Considering that the ultimate objective of the CONCERT framework as detailed in Chapter 3 is to support human mobility, i.e. to provide tourism information services based on the context of visitors, the intended users of the ontology are tourists, travellers and visitors.

However, visitors will not have to directly operate with the ontology, i.e. the information that they will receive in the mobile devices will have been processed by the ontology instead. In addition, the information that they need to provide to the CONCERT framework will not be provided directly to the ontology. There will be an application interface that will enable users to select some information from a given list, and it will be the application itself the one that will introduce this information in the ontology, as will be explained in Chapter 5.

The following list summarizes and enumerates the potential users:

- *User 1*: Visitors. They shall be the main users of this ontology. They will not manipulate the ontology themselves, but they ought to experience its functionality through the CONCERT-based application (Section 5.4)

- *User 2*: Tourism authorities and destination management organizations (DMOs). They can use this ontology to:

  - Provide content and publish tourism related information.
  - Measure and analyse tourists flows in order to support knowledge-based decision making processes.
  - Do marketing, e.g. DMOs or tourism companies can advertise themselves.

- *User 3*: Travel and tourism industries: to market their products in the Tourism Objects ontology (Section 4.6), so that they can be offered to visitors.

In spite of these different types of users having been identified, the type of users that this dissertation will focus on is “Visitors”. In fact, real visitors will accomplish the user evaluation carried out in Chapter 6.

Task 3: Identify intended uses

Again, considering the main objective of the CONCERT framework and having analysed the evaluation scenarios defined in Section 6.1, the following intended uses have been identified for the network of ontologies:

- *Use 1*: To model the context of people on the move considering characteristics typically related to the framework of human mobility.
- *Use 2*: To support semantic categorization of visitors.
- *Use 3*: To boost the use of semantic technologies to represent tourism related information and services. Tourism authorities and destination management organizations (DMO) may be interested in marketing their products in this ontology.
- *Use 4*: To foster a standard semantic representation of tourism objects.
- *Use 5*: To foster a *de facto* standard semantic model of context.
- *Use 6*: To provide tourism information based services on the move.
- *Use 7*: To support search of tourism objects.

In spite of these different types of potential uses having been identified, the type of uses considered in this dissertation will focus just on providing context-based tourism information services on the move. A rule-based engine built on top of the network of ontologies as will be specified in Section 5.4.3 will exploit the reasoning features of the network of ontologies.

Task 4: Identify Requirements

The set of requirements the network of ontologies needs to fulfil has been specified using the competency questions techniques[GF95]. They can be formulated by writing the questions in natural language by hand or several other tools such as Excel spread sheets, mind maps or collaborative tools.

There are a number of approaches to specify the competency questions:

- *Top-down*: Complex questions are decomposed in simple questions.

- *Bottom-up*: Simple questions are organised to form more complex questions.
- *Middle-up*: A combination of the first two approaches.

In the case of the ContOlogy network of ontologies a middle-up approach has been followed. The brainstorming process that has motivated the competency questions has been based upon the definition of the notion of context and its subsequent subdivision in different categories as shown in Section 3.1.2. A total of 65 competency questions have been identified and are listed below:

1. What kind of visitors are expected to use the platform?
2. How is a visitor identified?
3. What is the nationality of a visitor?
4. What kind of relationship may a visitor maintain at a certain moment with other humans?
5. What kind of activities may a visitor undertake in a time range?
6. What user parameters are necessary to define the activities?
7. What parameters of the visitor can be known at a certain moment of time?
8. What needs can a user have at a certain moment of time?
9. What kind of information/services can be provided to a visitor at a certain moment of time?
10. What kind of mobility patterns can be defined for a visitor? (Car, bus, on foot, and so on.)
11. Is there a context history of the visitor?
12. What is the objective of a visitor?
13. What is the motivation of the visitor?
14. What is the intention of the visitor?
15. What are the preferences of the visitor in terms of language?

16. What are the preferences of the visitor in terms of food?
17. What are the demographics of a visitor?
18. Is the visitor carrying a mobile electronic device?
19. What is the role of a visitor?
20. What is a visitor doing?
21. What activities are there?
22. What is an activity composed of?
23. What kind of device is it?
24. Where is the device located?
25. What are the device's characteristics?
26. How much battery has the device got left?
27. What kind of operating system does the device run?
28. Which information networks are accessible for a device?
29. What kind of screen does the device have?
30. Which resources have got a certain network?
31. Whose operator is this network?
32. Which are the networks detected by an electronic mobile device at a certain moment of time?
33. What is the band-width the network is providing?
34. Does network X provide a particular band width Y?
35. What networks detected by an electronic mobile device are free?
36. Which networks have the largest band width?
37. Which are the coordinates in which the visitor is located at a given moment of time?
38. Is the visitor at the considered moment located somewhere?

39. Which points of interest are nearby the specified location?
40. Is it about an indoor or an outdoor location?
41. Which is the location's accuracy?
42. Which street, city or region does the location correspond to?
43. In which street is a particular building?
44. Which is the last location known for the particular traveller?
45. What time is it?
46. What day is it?
47. What month is it?
48. What day of the week is it?
49. Is it earlier the 5 o'clock p.m.?
50. Is it later that 12 o'clock p.m.?
51. What is the weather like today?
52. Is it raining?
53. Is it cold?
54. Is it hot?
55. What is the temperature outside?
56. Is it cloudy?
57. What kind of tourism services are offered in the environment?
58. What kind of restaurants are offered in the area?
59. Is a certain restaurant vegetarian?
60. Does a restaurant provide the service in a certain language?
61. Where is the restaurant located?
62. How much does a restaurant cost?
63. Does a restaurant provide outdoor services?

64. Is there any point of interest nearby?

65. Is there any accommodation facility around?

The following picture shows an excerpt of some of the competency questions, during the brainstorming sessions.

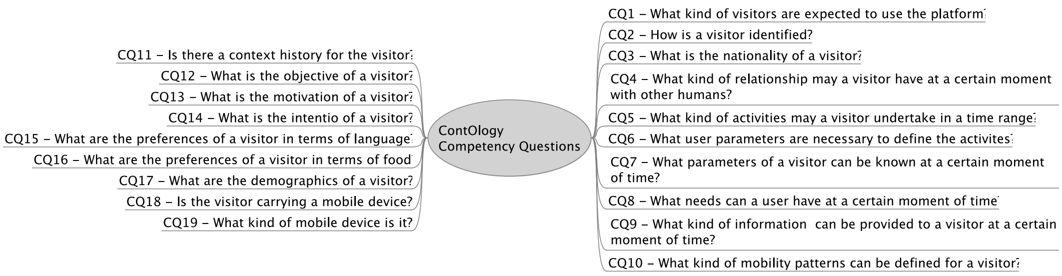

Figure 4.3: Excerpt of Competency Questions in a brainstorming session

Task 5: Group requirements

The objective of this task is to group the competency questions into several categories. The output of this task is a set of groups, i.e. categories, that include a set of the competency questions.

This way, the 62 competency questions have been manually grouped into 7 groups as shown in figure 4.4. The competency questions have been grouped in a way that each of the group includes questions that are relevant only to a specific feature of the ontology. In order to determine the identity of the groups and the group to which each of the competency question belongs, the following approaches can be followed:

- Either analyse the frequency of terms and later group the competency questions based on terms that have a higher frequency.

- Or use pre-established categories such as visitor, location, time, etc.

The classification of the competency questions into groups or categories has not been complex, since the competency questions have been actually approached from the different categories of context constituents identified in the definition of context presented in Section 3.1.2. The Competency questions described in Task 4 have been manually grouped into six groups as follows:

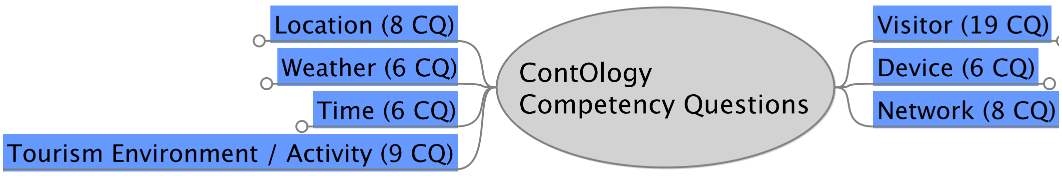

Figure 4.4: Grouping of the CQ

- Visitor.
- Device.
- Network.
- Location.
- Weather.
- Time.
- Tourism Environment / activity.

Task 6: Validate the set of requirements

The goal of this task is to identify possible mistakes, conflicts between competency questions, missing competency questions and contradictions. The output of this task is a confirmation about the validity of the competency questions.

The NeOn methodology for building networks of ontologies specifies a number of criteria to validate the requirements defined with the set of competency questions. These requirements are correctness, completeness, consistency, verifiability, understandability, no ambiguity, conciseness, realism, modifiability and traceability.

During the overall process of the ContOlogy network of ontologies specification, the following issues were considered in order to guarantee its correctness and validity:

- *Correctness*: The set of requirements is correct if each requirement refers to some features of the ontology to be developed, i.e. the formulation of the competency questions needs to be correct and needs to address a certain feature of ContOlogy. The formulation and answers of the competency questions have been proven to be correct.

- *Consistency*: The set of requirements can be considered correct if there are no conflicts between them. The analysis of the set of competency questions listed previously has proven to be consistent.
- *Verifiability*: A set of requirements is verifiable if and only if each requirement is verifiable. The final ontology satisfies each competency question.
- *Understandability*: A set of competency questions must be understandable by users. The set of competency questions listed previously have been understood.
- *Unambiguity*: Given that each of the competency questions has been considered consistent and correct, no room has been found for ambiguity in their formulation.
- *Conciness*: Each requirement is relevant and no duplication has been identified.
- *Realism*: Requirement meanings must make sense in the domain.
- *Modifiability*: A set of requirements is modifiable if its structure and style allow to change issues in an easy, complete and consistent way. In the ContOlogy network of ontologies, the set of competency questions have been formulated from a structured point of view, i.e. it could potentially be modified.
- *Traceable*: The origin of all competency questions can be identified at any moment.

There is only one criterion, i.e. Completeness, that has not been included in the validation task. This is due on the one hand, to the broad scope of the notion of context and on the other, to the fact that no external sensors are used to gather context information. Even though the scope of the notion of context has been clearly identified and its scope set within this dissertation, the way in which context information is going to be gathered (without sensors) poses some limitations on the following matters: First, the kind of context information that may be gathered. Second, on its completeness. This, on the contrary, does not mean that the CONCERT framework will fail to provide context based information, rather it refers to the fact that particular types of information will not be taken into account. The lack of completeness has also been reported in other cases of use of ontologies [VFC05] [JMSK09] [AQFHS07].

For example, not using sensors to gather context information would prevent the CONCERT context model of measuring the degree of lightning, or the amount of noise at a particular location. This information not being particularly relevant to support human mobility, does not allow to consider the context model complete from an objective point of view.

Task 7: Prioritize requirements

The goal of this task is to provide different levels of relevancy to the different groups of competency questions (Task 5) and to each question identified within each of the groups. The output of this task is a set of priorities attached to each group of competency questions and to each question inside a competency question group.

In this case, no priorities have been established for the group identified, meaning that all of them have been considered in the different iterations (see Section 4.5.4, where the number of iterations are established according to the ontology life cycle) of the network of ontologies building process. However, not all questions within the groups have been given the same relevance and have been considered at different stages of the building process.

Task 8: Extract terminology frequency

The goal of this task is to extract from the list of competency questions a pre-glossary to be used in the conceptualization activity. Taking the list of competency questions presented earlier and their answers, the mostly used terms shall be obtained, see table 4.2.

The terminology (names, adjectives, and verbs) can be extracted from the competency questions that will formally be represented in the network of ontologies by means of concepts, attributes and relations.

From the answers to the competency questions, the objects of the universe of discourse that will be represented as instances can be determined. For example, the answer to *who is the visitor?* could be Carlos, therefore *Carlos* would be one possible instance of the class Visitor.

Term	Frequency
Visitor	16
User	2
Language	2
Device	9
Location	5
Network	8
Weather	1
Time	5
Tourism Objects	9

Table 4.2: Terminology and Frequency

4.5.3 Select processes and activities

The goal of this task is to select the set of processes and activities that need to be performed during the development of the network of ontologies. The NeOn methodology establishes the relationship between different scenarios, processes and activities that have to be carried out within the general process of ontology development[SFdCB+07].

The scenarios for building the ContOlogy network of ontologies have already been chosen in Section 4.5.1. Therefore, the list of processes and activities required in the development can be quite straight forwardly obtained from the following table.

Table 4.3 shows the processes and activities considered under each of the scenarios in the following way: (i) a process or an activity may be carried out in a particular scenario (S1); (ii) a process or an activity can be carried out in a list of scenarios (S4, S6, S8); or (iii) a process or an activity can be carried out in a set of scenarios (S1-S9).

Tables 4.3 and 4.4 show, through text indent, which activities belong to which process:

Name of the Process (P) or activity	**Related Scenarios**
Ontology Annotation	S1 - S9
Ontology Assessment	S1 - S9
Ontology Comparison	S3 - S4
Ontology Configuration Management	S1 - S9
Ontology Conceptualization	S1
Control	S1 - S9
Ontology Customization	S8
Ontology Diagnosis	S1 - S9
Ontology Documentation	S1 - S9
Ontology Elicitation	S1 - S9
Ontology Enrichment	S4, S6, S8
ontology environment study	S1
Ontology Evaluation	S1 - S9
Ontology Evolution	S1
Ontology Extension	S4, S6, S8
ontology Feasibility Study	S1
Ontology Formalization	S1
Ontology Forward Engineering	S2, S4, S6
Ontology Implementation	S1
Ontology Integration	S2 - S6
Knowledge Acquisition for ontologies	S1 - S9
Ontology learning	S1 - S9
Ontology modification	S1
Ontology Modularization	S4, S6, S8
Ontology Module Extraction	S4, S6, S8
Ontology Partitioning	S4, S6, S8
Ontology Population	S1 - S9
Ontology Pruning	S4, S6, S8
Ontology Quality Assurance	S1 - S9
Ontology Reengineering	S4, S6
Ontology Restructuring	S4, S6, S8
Ontology Repair	S1 - S9
Ontology Re-use	S3 - S6

Table 4.3: Selected processes and activities (I)

Name of the Process (P) or activity	Related Scenarios
Ontology Statement Re-use	S3 - S6
Ontology Design Pattern Re-use	S7
Ontology Reverse Engineering	S4
Ontology Scheduling	S1
Ontology Search	S3 - S6
Ontology Selection	S3 - S6
Ontology Specialization	S4, S6, S8
Ontology Specification	S1
Ontology Summarization	S1 - S9
Ontology Translation	S1
Ontology update	S1
Ontology upgrade	S1
Ontology Validation	S1 - S9
Ontology Verification	S1 - S9
Ontology Versioning	S1

Table 4.4: Selected processes and activities (II)

These activities and processes are the ones that need to be carried out according to the Scenarios of ontology development set previously for the development of the ContOlogy network of ontologies. However, in order to facilitate the identification of the activities that are important and necessary when developing software products, some IEEE standards identify which activities are required and optional[IEE96, IEE97b, IEE97a].

Activities within SW development are categorized either as mandatory or as "if applicable"[IEE97b, IEE97a]. Evidently, mandatory activities must always be carried out. On the contrary, there are some explanations of the cases in which "if applicable" activities may not be carried out.

According to the previously stated, the following list gathers the processes and activities that will be carried out in order to develop the ContOlogy context ontology.

- Knowledge acquisition for ontologies
  - Ontology Elicitation
  - Ontology Population
- Ontology Annotation
- Ontology Conceptualization

- Control
- Ontology Documentation
- Ontology Evaluation
  - Ontology Validation (P)
    - Ontology Diagnosis
    - Ontology Repair
  - Ontology Verification (P)
- Ontology Evolution
- Ontology Formalization
- Ontology Quality Assurance
- Ontology Configuration Management
- Ontology Implementation
- Ontology Improvement
- Scheduling
- Ontology Reengineering (P)
  - Ontology forward reengineering
  - Ontology reverse reengineering
  - Ontology restructuring
    - Ontology Enrichment
    - Ontology Specialization
    - Ontology Extension
  - Ontology Modularization
    - Ontology Division
  - Ontology Pruning
- Ontology Summary
- Ontology Re-use
  - Ontology Search

  - Ontology Comparison
  - Ontology Integration
  - Ontology Selection

- Ontology design patterns re-use (P)
- Ontology Assessment
- Ontology Versioning

4.5.4 Plan and Select the Life Cycle of ContOlogy

Once the scenarios have been identified, activities and processes specified, the latter has to be related to the former. The relation between activities, processes and scenarios has been shown in Table 4.3.

The requirements of the network of ontologies have also been identified in Section 4.5.2, therefore, the next step within the ontology building process is to plan and schedule the tasks to be performed within the development process.

The planning activity identifies the different processes and activities that are going to be carried out during the ontology building process and organizes them according to a time scale[SFFLGP+08][SFGP08]. Like in any normal software development project, this planning shall guide the development of the network of ontologies including further information on the sequence of activities and restrictions in terms of time and human resources.

The NeOn methodology considers five tasks to accomplish the planning activity[SFFLGP+08]:

- Select the ontology network life cycle model.
- Select processes and activities.
- Map the processes and activities in the ontology network life cycle model.
- Establish the sequence of processes and activities.
- Establish time and human resources restrictions.

An ontology life cycle model[SFdCB+07] determines how to develop an ontology network project, i.e. how to develop and maintain a network of ontologies, i.e. how to organize the ontology activities into phases or stages.

In general, life cycle models can be seen as abstractions of the phases or stages through which a product passes along its life[SFFLGP+08]; in the case of this dissertation, the product is the ContOlogy network of ontologies that models and manages the context of visitors. Life cycle models are used to represent the entire life of a product from concept to disposal and they determine the order of the phases and establish the transition criteria between different phases.

The goal of this task is to choose the most appropriate ontology network life cycle model for ContOlogy[SFFLGP+08]. To help make this decision the next set of natural language questions are proposed:

- Are the requirements of the network of ontologies assumed to be fully known at the beginning of the development process?
  - If yes, a waterfall ontology network life cycle can be chosen.
- Are the requirements of the network of ontologies likely not to be fully known at the beginning of the development process? And/or, are they likely to change during the development process?
  - If yes, the iterative-incremental network of ontologies life cycle is more appropriate.
- Have the requirements of the network of ontologies some priorities assigned?
  - If yes, the iterative-incremental ontology network life cycle is more appropriate.

In the case of the ContOlogy network of ontologies, the requirements of the network are not fully known at the beginning of the building process. The definition of context put forward earlier, the implications of establishing a double level of interoperability (both at the infrastructure level and at the model level) and using digital broadcasting for context and tourism information dissemination provide some potential requirements the network of ontologies ought to meet. However, just considering these requirements would be far from being complete. Therefore, the final requirements the network of ontologies needs to fulfil have to be further analysed by more deeply working on the competency questions and in additional brainstorming sessions.

Due to these reasons, the answer to the first question of the previous set is “no”, since the requirements are not fully known at the beginning. Therefore, the answer to the second and third questions is “yes”. This applied in terms of the NeOn methodology means that the life cycle of the network of ontologies is an iterative-incremental life cycle model. Again, according to the NeOn methodology, should the iterative-incremental ontology life cycle be selected, the type of waterfall model[SFFLGP+08] to be used in each of the iterations has to be selected following.

Experience shows that the ontology building process is iterative. Therefore, it is necessary to determine at this point the number of iterations the process will comprehend. The proposed number of iterations is three: The first ontology will be determined on basis of the definition of the notion of context put forward, the architecture of the system (Section 5.1) and objectives of the platform. These requirements will be enriched with a set of competency questions.

The second iteration will consider the evaluation of the first iteration and will take into account a brainstorming session.

Finally, the third iteration will consider the evaluation of the second iteration and a second brainstorming session. Considering aspects of human behaviour, decision making processes, etc. a survey will be designed and its results after asking a representative sample of travellers will be assessed and incorporated into the requirements, together with definitions provided by the UNWTO[UNW08a].

In order to determine the waterfall model to use within each iteration, the following set of natural language questions needs to be answered:

- Is it planned or said somewhere in the ontology requirements that already existing ontological resources ought to be used?
  - If yes, the five-phase waterfall model should be elected;
- Is it planned or said somewhere in the ontology requirements that ontology design patterns ought to be used in the ontology network?
  - If yes, then the five-phase waterfall model ought to be chosen.
- Is it planned or said somewhere in the ontology requirements to use and merge a set of already existing ontological resources in the ontology network?
  - If yes, the five-phase + merging phase model ought to be selected.
- Is it planned or said anywhere in the ontology requirements to use any non-ontological resources such as thesauri, databases, etc., in the ontology?

  - If yes, the six-phase model ought to be used

- Is it planned or said somewhere in the ontology requirements to use and modify any existing ontological resource in the ontology network?
  - If yes, the six-phase model should be used;

- Is it planned or said somewhere in the ontology requirement to use, merge and modify a set of existing ontological resources in the ontology network?
  - If yes, the six-phase + merging phase model ought to be followed.

The answers to these questions have been summarized in the following table:

	Iteration		
Questions/Associated Life-Cycle Model	First	Second	Third
Is it planned or said somewhere in the ontology requirements that already existing ontological resources ought to be used?	Yes	Yes	Yes
Is it planned or said somewhere in the ontology requirements that ontology design patterns ought to be used in the ontology network?	Yes	Yes	Yes
Is it planned or said somewhere in the ontology requirements to use and merge a set of already existing ontological resources in the ontology network?	No	No	No
Is it planned or said somewhere in the ontology requirements to use any non-ontological resources such as thesauri, databases, etc. in the ontology?	No	No	No
Is it planned or said somewhere in the ontology requirements to use and modify any existing ontological resource in the ontology network?	Yes	Yes	Yes
Is it planned or said somewhere in the ontology requirement to use, merge and modify a set of existing ontological resources in the ontology network?	No	No	No

Table 4.5: Relation between scenarios and life cycles

According to the answers provided, the following waterfall life cycle models for each of the iterations within the iterative-incremental cycle model may be followed to develop the network of ontologies:

- Five-phase waterfall life cycle model.
- Six-phase waterfall life cycle model.

Theoretically speaking, any of these two life cycle models could be followed. However, for each of the iterations the most specific life cycle model ought to be elected. This decision can be taken by the use of the following pyramid (see Table 4.6), which represents the versions of the waterfall life cycle put forward[SFFLGP+08].

<table>
<tr><td colspan="2">Six-phase model + mix</td></tr>
<tr><td>Six-phase model</td><td>Five-phase model + mix</td></tr>
<tr><td colspan="2">Five phase model</td></tr>
<tr><td colspan="2">Four phase model</td></tr>
</table>

Table 4.6: The waterfall pyramid

As the five-phase waterfall life cycle model grounds the six-phase model, the six-phase waterfall model shall be used in the building process of the ContOlogy network of ontologies.

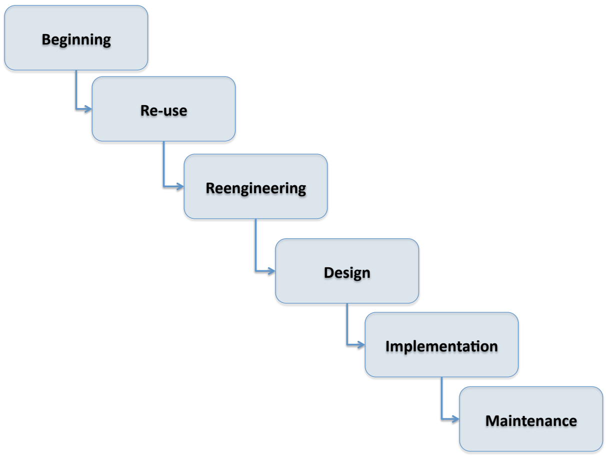

Figure 4.5: The Waterfall life cycle model

4.5.5 Map Processes and activities into the life cycle model

The objective of this task is to set the correspondence between processes and activities and the life cycle model of the network of ontologies. The kind of ontology network life cycle has been identified in the previous task. According to the characteristics of the network of ontologies to implement and to the answers given to the questions set by the NeOn methodology to build networks of ontologies, it was determined that the building process would be a 3 iteration iterative-incremental six-phase waterfall life cycle model.

Corresponding phase	Name of process or activity
Initiation phase	Scheduling
Re-use phase	Ontology Search Ontology Comparison Ontology re-use Ontology design pattern re-use Ontology selection
Reengineering phase	Ontology division Ontology enrichment Ontology Specialization Ontology Extension Ontology forward engineering Ontology reverse engineering Ontology modularization Ontology pruning Ontology restructuring Ontology reengineering
Design phase	Ontology conceptualization Ontology Evolution Ontology Formalization Ontology Integration
Implementation phase	Ontology implementation
Maintenance phase	Ontology improvement Ontology versioning
All phases	Knowledge Acquisition for ontologies Ontology Annotation Control Ontology Diagnosis Ontology Documentation Ontology Quality Assurance Ontology Configuration Management Ontology Population Ontology Repair Ontology Summary Ontology Validation Ontology Assessment Ontology Verification

Table 4.7: Mapping Processes and Activities to Phases in the Waterfall Life Cycle

So, in order to determine the activity that has to be carried out in each phase or phases of the ontology life cycle model, the NeOn methodology establishes a table4.7 that relates processes and activities with phases of the life cycle model[SFFLGP+08].

4.5.6 Set the sequence of processes and activities

Once the correspondence between activities and life cycle model has been set, the sequence in which these activities are going to be accomplished has to be arranged. The result of this will be the entire life cycle of the network of ontologies.

The order in which processes and activities will be undertaken is determined according to three important factors:

1. The initial sequence of processes and activities will be decided on the life cycle model of choice.
2. Some of the activities and processes can be performed in parallel rather than in series.
3. The activity input output criteria may influence the sequence in which they are carried out. Output information availability of a process or an activity may determine the beginning of another process or activity. This is due to the fact that the second process or activity may require some input information that it is output information at the same time of the first process or activity.

According to all of the above, the following is the sequence of the tasks in each of the iterations:

- Iteration 1
  - Initiation
    - Scheduling
  - Re-use phase
    - Ontology re-use
    - Ontology design patterns re-use
  - Reengineering phase
    - Ontology reengineering
  - Design phase

    - Ontology conceptualization, formalization
    - Ontology evolution
  - Implementation phase
    - Ontology implementation
  - Maintenance
    - Ontology improvement
    - Ontology versioning

- Iteration 2
  - Initiation
    - Scheduling
  - Re-use phase
    - Ontology re-use
    - Ontology design patterns re-use
  - Reengineering phase
    - Ontology reengineering
  - Design phase
    - Ontology conceptualization, formalization
    - Ontology evolution
  - Implementation phase
    - Ontology implementation
  - Maintenance
    - Ontology improvement
    - Ontology versioning

- Iteration 3
  - Initiation
    - Scheduling
  - Re-use phase
    - Ontology re-use
    - Ontology design patterns re-use
  - Reengineering phase

    * Ontology reengineering
  - Design phase
    * Ontology conceptualization, formalization
    * Ontology evolution
  - Implementation phase
    * Ontology implementation
  - Maintenance
    * Ontology improvement
    * Ontology versioning

Tasks to be performed at all time in parallel are the following:

- Knowledge acquisition for ontologies.
- Ontology annotation.
- Control.
- Ontology Documentation.
- Ontology Evaluation.
- Ontology quality assurance.
- Ontology configuration management.
- Ontology summary.
- Ontology assessment.

4.5.7 Establish the resource restrictions and assignments

Since the development of the network of ontologies has been carried out in the framework of this dissertation, this phase does not apply for the purpose of this work. There are neither resource restrictions nor assignments of the work to be done.

4.6 ContOlogy: Description of the network of ontologies

Once the requirements of the ContOlogy network of ontologies have been defined in Section 4.5.2 and the implementation of the ontologies has been scheduled, the next step consists of implementing the ontologies. The domain covered by the ContOlogy network of ontologies is that of visitors' context, as stated earlier. Due to the complexity of such a domain on the one hand, and to the existence of some ontologies that represent and model context (in a general way and to some extent) that can be (at least) partly re-used on the other, together with the fact that the resulting context model needs to be as modular, flexible and scalable as possible, a network of ontologies has been developed following the definition of context put forward in Chapter 3.

Using a network of ontologies will allow the model to cover the notion of context in a more complete way and in addition, its modularity, scalability and re-usability will be enhanced. Both the literature and experience have shown that the ontology building process is iterative[NM01]. Thus, as stated throughout Section 4.5, the ContOlogy context network of ontologies will be implemented in three consecutive iterations, each of them providing a working prototype of the ontology network suitable for validation of the model. For operative reasons, the number of iterations has been set in three, however, the number of iterations could go on as long as there is time and resources to work on the ontology development.

At the end of building process, the network of ontologies integrates at the moment 11 ontologies. All together there are 86 classes, 41 object properties, 22 datatype properties, 43 restrictions and 73 instances or individuals. The following table shows the ontologies that configure the network with a short description about the entity the ontology represents (the ontology is thoroughly analysed in Section 4.7). The language used to specify each of the ontologies has been OWL in its DL sublanguage. The level of expressivity shown by the network of ontologies is SHOIN(D), which supported by the OWL DL.

The components of the network of ontologies, i.e. context constituents, have been determined considering the definition of the notion of context put forward earlier, the CONCERT architecture that is thoroughly defined in Chapter 5 and the competency questions defined in Section 4.5.4. In addition to that, the objective pursued by the CONCERT framework has allowed to establish the relationships amongst the ontologies of the network.

Figure 4.6 shows a high level overview of the ContOlogy network of ontologies.

Ontology	Definition
Visitor	Characteristics of the human being *en route* [UNW08a].
Preferences	Information that describes visitors' personal characteristics, demographics, and so on[UNW08a].
Role	The role that visitors play at a certain moment[UNW08a].
Motivation	Represents the reason why the visitor is travelling[UNW08a].
Activity	Represents what the visitor is doing at a certain moment of time. This information can be obtained in various ways, e.g. from the mobile device's agenda[PdBW+06].
Environment	Represents the surroundings of the visitor as well as the weather conditions at the location of visitors[PdBW+06].
Device	Physical object that visitors carry with them and that enables interaction with information[W3C04a].
Network	Infrastructure to connect devices and convey information[CRL+09].
Location	Coordinates that define where a visitor is at a given moment of time[PVR+06].
Time	Physical dimension that measures the span between facts[HP06].
Tourism objects	Represents the services provided in a certain environment[BFP+09].

Table 4.8: Ontologies of the ContOlogy network of ontologies

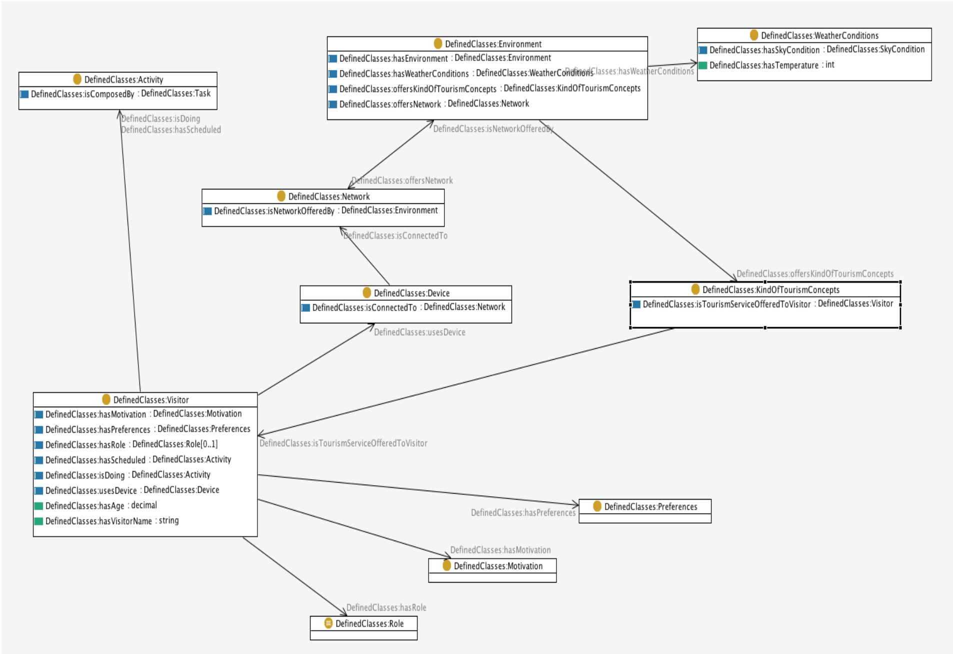

Figure 4.6: High level view of the ContOlogy network of ontologies

4.7 ContOlogy: Validation

This section is devoted to the validation of the ContOlogy network of ontologies. The validation has been performed following the guidelines provided by the NeOn methodology and assesses the extent to which the requirements established at the beginning of the process have been met by the resulting prototype according to the competency questions. In other words, if the ontologies of the network can provide answers to the competency questions set in Section 4.5.4, then the ontology can be considered valid for the purposes for which it was conceived.

The evaluation of each of the ontologies can be found below:

1. *Visitor*: In order to model the category of visitor, the recommendations established by the UNWTO to define visitors have been followed[UNW08a]. This part of the network of ontologies has been developed from scratch, since no ontologies have been found that implement the recommendations provided by the UNWTO. The ontology corresponding to the Visitor (figure 4.7) constitutes the

centre of the ContOlogy network of ontologies and is related to other ontologies, which have also been implemented following the UNWTO guidelines, as it is shown in the picture. This ontology has been positively validated against competency questions 1 to 10 (see page 111).

The *Visitor* ontology is related to the *Motivation, Role, Preferences, Activity* and *Device* ontologies through *Object properties*, i.e. the individuals of *Visitor* have relations with individuals of the aforementioned classes. These relations allow to link one particular individual of the class *Visitor*, i.e. a particular person, with its particular preferences, roles, and so on in the following way:

- *Visitor* A *hasRole Role* R.
- *Visitor* A *hasMotivation Motivation* M.
- *Visitor* A *hasPreferences Preferences* P.
- *Visitor* A *usesDevice Device* D.
- *Visitor* A *isDoing Activity* Z.

The *Visitor* ontology and its relations with other ontologies, especially with the *Role, Motivation* and *Preferences* ontologies, is crucial in order to define a person *en route*. In fact, the relations, i.e. the individuals of two different classes that are connected through the object property are the values that are taken into account by the rules to filter the incoming information.

2. *Preferences*: Preferences have been modelled following the UNWTO guidelines to define a visitors preferences[UNW08a]. Despite the fact that there are a number of ontologies that consider human preferences, the one implemented in the framework of this dissertation has been developed from scratch, since the preferences as defined by the UNWTO were not contemplated in the existing ontologies. This is also an important ontology, in fact its relation with the *Visitor* ontology can be seen in figure 4.8, since it represents the preferences of people *en route*. The validity of this ontology has been assessed through CQs 15 to 18.

   The *Preferences* ontology considers the following concepts (graphically available in figure 4.8:

   - Ability

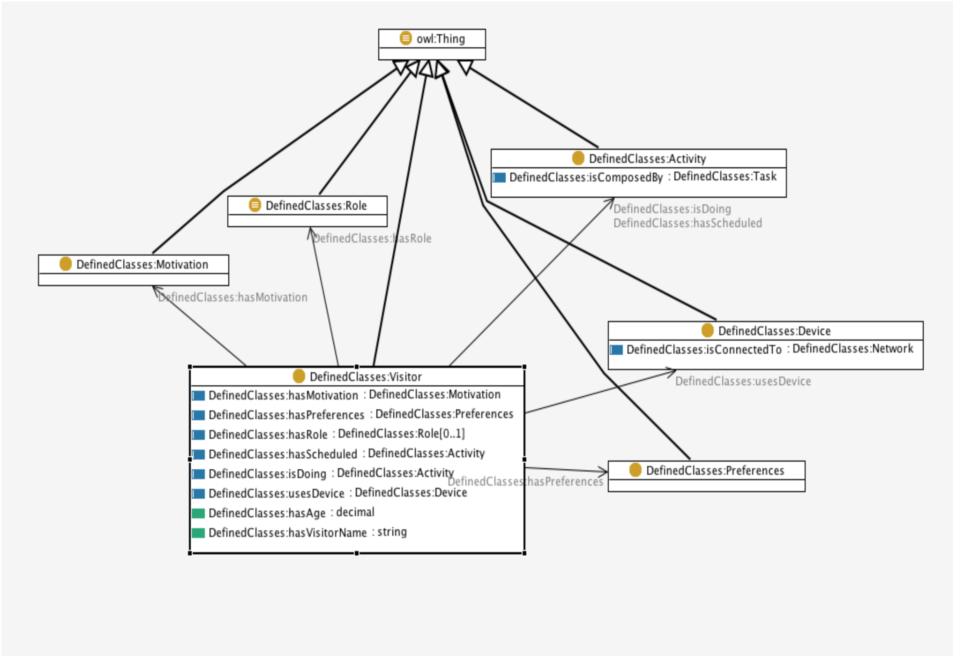

Figure 4.7: The visitor ontology and its relations with other ontologies

- Demographics
  - Age Range
  - Educational Level
  - Food Preferences
  - Gender
  - Language

For the purposes of the work carried out in this dissertation (see Section 6.2) the class *Demographics* has been extensively used, as it provides important information related to the likes and dislikes of visitors, critical for the filtering process. Figure 4.8 also shows the *Visitor* ontology and its relation to the *Preference* ontology via the *hasPreferecenes* object property.

3. *Motivation*: Motivation has been modelled following the UNWTO guidelines to define visitors roles. This ontology (figure 4.9) is structured and includes concepts such as the ones below:

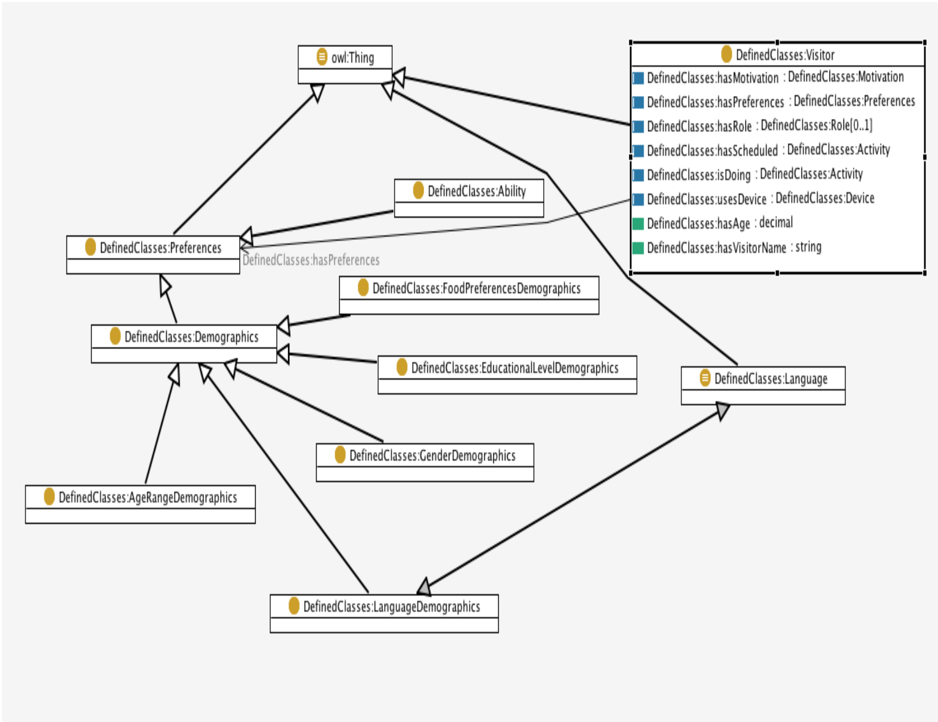

Figure 4.8: The preferences ontology and its relations with other ontologies

- Business motivation
  - Attending conference
  - Attending meeting
  - Attending fair
  - Attending congress
- Personal motivation
  - Holiday leisure
  - Religion pilgrimage
  - Education training
  - Health medical
  - Visiting friends and relatives
  - Shopping
  - Transit
  - Other kinds of motivations

This ontology represents another core ontology of the ContOlogy network of ontologies. It comprehends all the travelling motivations defined by the UNWTO. The ontology has been developed from scratch. There are ontologies that model motivations, however, none of them have been developed following the UNWTO premises and therefore, do not address the requirements of human mobility in terms of motivations. This ontology has been validated against CQs 12 to 14 and its details of the ontology can be seen in figure 4.9. The figure shows the *Motivation* ontology, its two subclasses, namely *BusinessMotivation* and *PersonalMotivation* and each of their individuals.

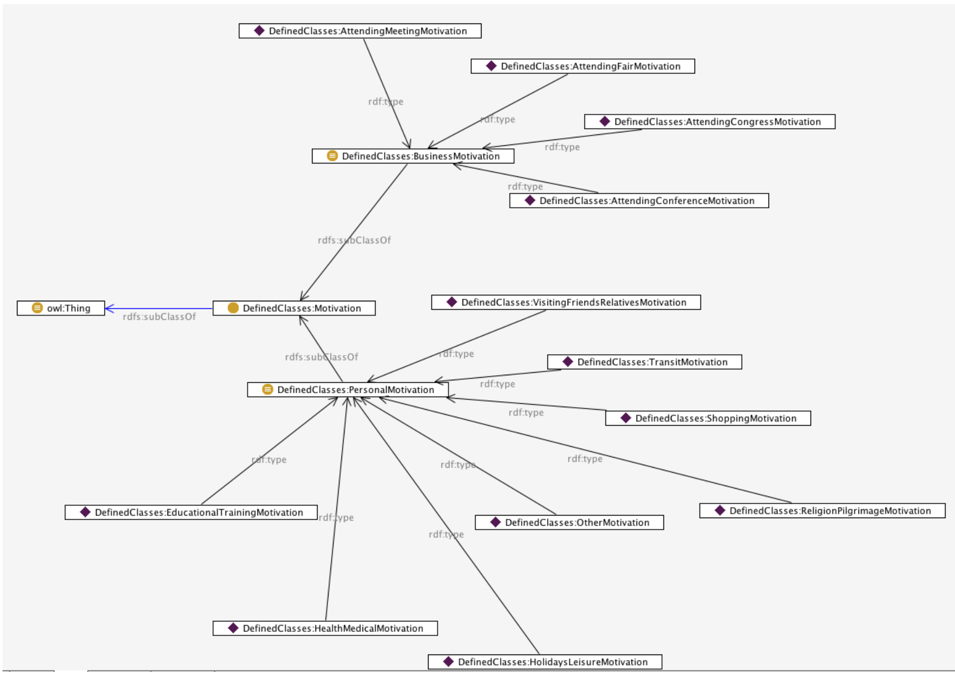

Figure 4.9: The motivation ontology and its relations with other ontologies

This ontology is related to the *Visitor* ontology through the *hasMotivation* object property.

4. *Role*: Role (figure 4.10) has been defined in relation with the activity a visitor is undertaking at a given moment of time and as to provide multiple possibilities in service delivery. This ontology has also been defined according to the guidelines provided by the UNWTO and has

been developed from scratch. The *Role* ontology has been assessed against the CQ 19. For example, one may be a visiting student at a university, therefore having an educational training kind of motivation, but be at the same time be attending a meeting, therefore, playing the role of attending meeting visitor at that given moment of time.

This ontology is defined by a single "defined" class, i.e. a class that only contains individuals, which happen to be the different roles defined by the UNWTO and that one individual can assume at a certain moment depending the activity it may be doing.

This being so, the kind of services this particular individual would require would be different from the ones required by somebody having an educational training kind of motivation but also being on holidays for a weekend. The potential combination of roles and motivations allows to expand the range of services that can be offered to visitors.

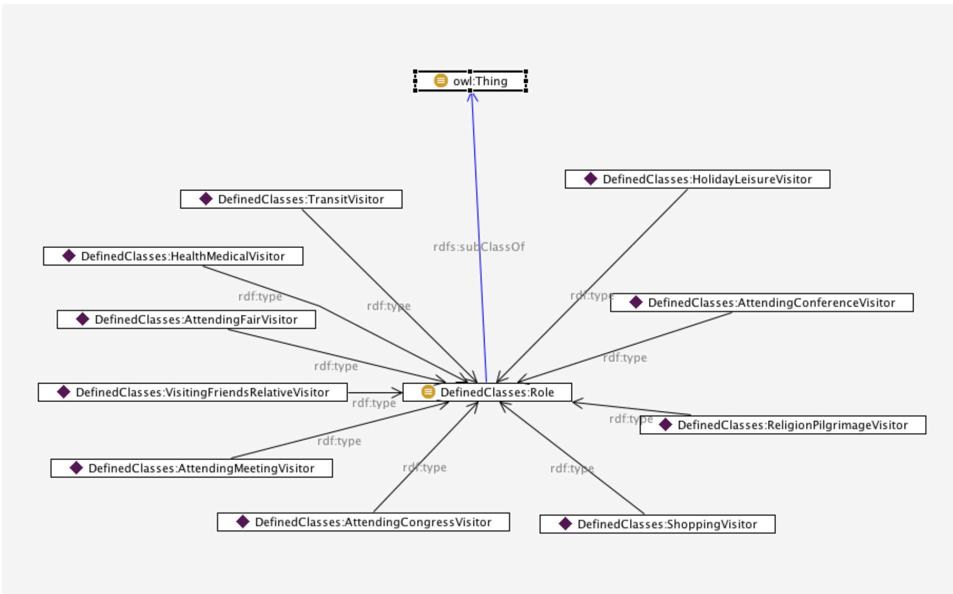

Figure 4.10: The role ontology and its relations with other ontologies

5. *Activity*: The activity of a visitor at a given moment of time represents what the visitor is doing. This ontology (figure 4.11) has been imported

from the Comanto ontology[SRA06], also revised in Section 2.5. This is an important ontology within the ContOlogy network of ontologies for the its relation with the *Role* ontology. The ontology has one subclass, namely Task. The relation of *Task* and *Activity* is that an activity may be composed of one or more tasks. These two classes are related through an object property, namely *Activity isComposedBy Task*.

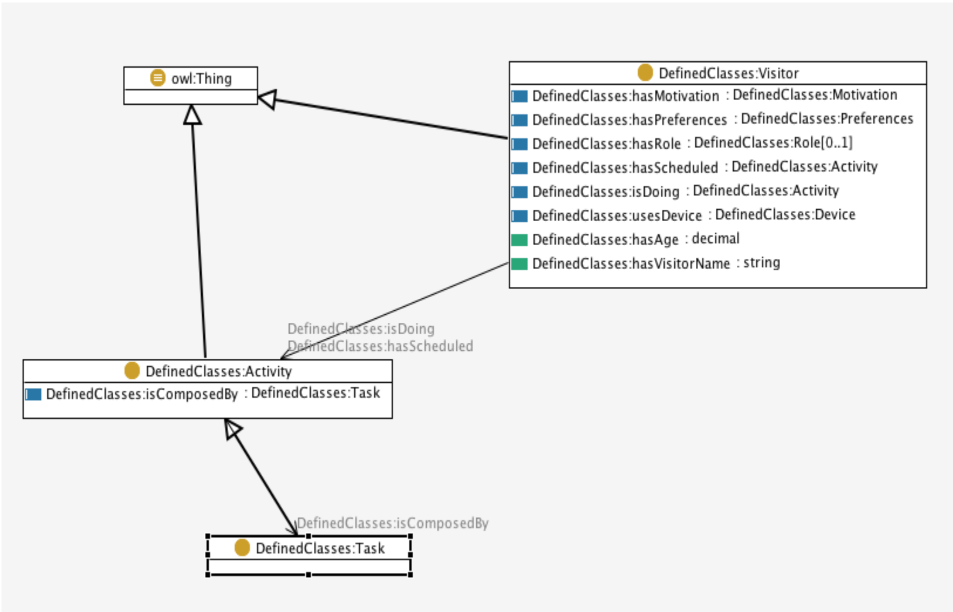

Figure 4.11: The activity ontology and its relations with other ontologies

However, since the experiments performed in this dissertation have not been to tightly related with the kind of activity, this ontology has plenty of room to be modified in the future. It has been assessed against CQ 5, 20, 21 and 22.

6. *Device*: This ontology (figure 4.12) represents the physical object that visitors carry with and allow them to be permanently connected to sources of information[W3C04a]. This ontology is made up of the following concepts:

   - Terminal browser

- Terminal hardware
  - Device hardware platform
- Terminal software
  - Device software platform
- Terminal type

An existing ontology has been partly re-used in order to build the *Device* ontology[W3C04a]. The terms shown in the previous list have been taken from the CC/PP ontology and an object property between *Visitor* and *Device* has been defined, namely *usesDevice*, in order to link the two ontologies, thus assigning a particular device to a particular visitor. This ontology has been validated against CQs 23 - 29.

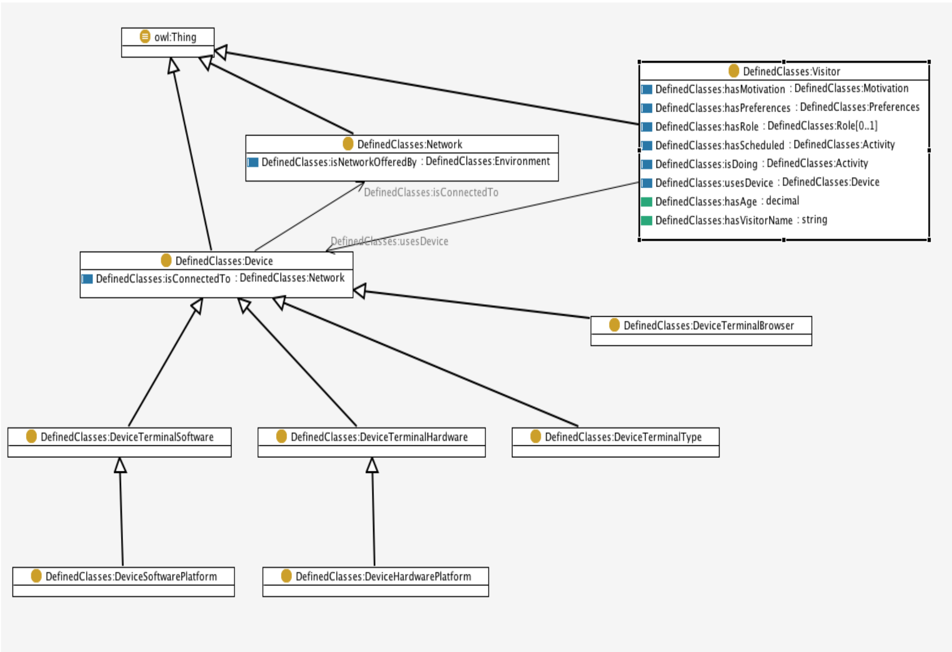

Figure 4.12: The device ontology and its relations with other ontologies

7. *Network*: This ontology (figure 4.13) represents the infrastructure that provides the means of communication and to which mobile devices are connected in order to receive information[CRL+09]. This ontology

has been partially imported from the mIO! ontology[CRL+09]. The ontology is further subdivided in the following concepts:

- Network bandwidth
- Network operator
- Network resources
- Network usage cost

This ontology is related to the *Device* ontology via an object property called *isConnectedToNetwork*. This is an object property that relates and individual of the *Device* ontology, i.e. a particular device, with another individual of the *Network* ontology, i.e. a particular network, e.g. CICtourGUNE Wi-Fi network. The details of this ontology can be examined in figure 4.13. This ontology has been validated against CQs 30 - 36.

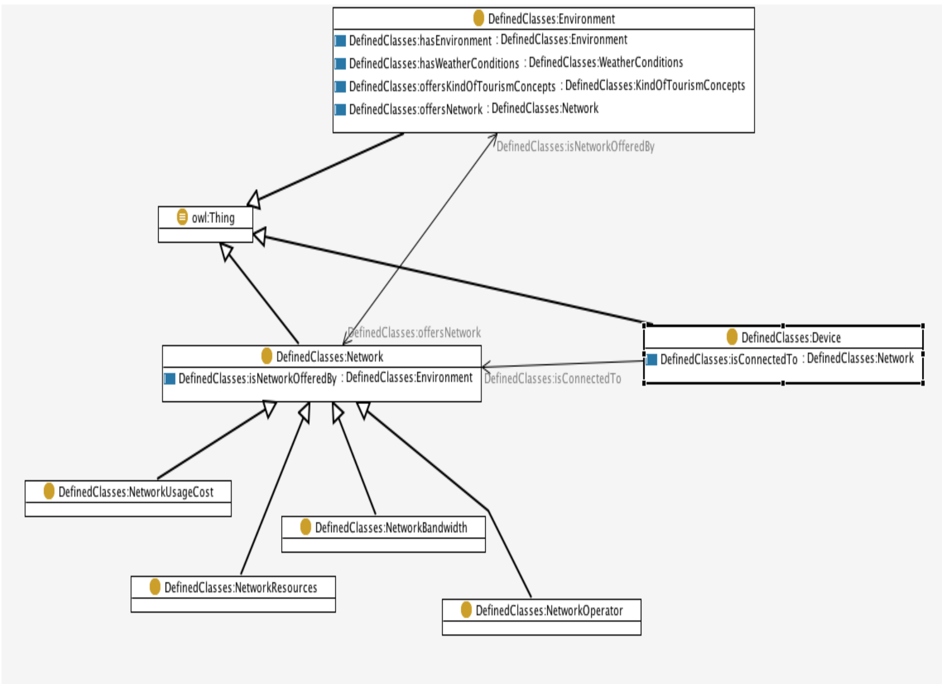

Figure 4.13: The network ontology and its relations with other ontologies

8. *Environment*: This ontology (figure 4.14) represents the surroundings of a visitor and has been defined as in the CoDaMoS ontology[PVR+06]. It is further subdivided in the following concepts:

    - Area
    - City
    - Neighbourhood

    Figure 4.14 represents graphically the ontology. The *Environment* ontology is linked to other ontologies in the network as follows:

    - It is related to the *Network* ontology via the *offersNetwork* object type property.
    - It is related to the Weather ontology through the *hasWeatherConditions* object property.
    - It is related to the *KindOfTourismConcepts* through the *object property.*

    This means that through the relations established in the network of ontologies, a visitor uses a device, which is connected to a network, which is in a certain environment that has a certain location. Environment and location are related through the *hasLocation* object property, which has an open domain and ranges the *Environment* ontology. This ontology has been validated through CQ 57.

9. *Tourism Objects*: This ontology (figure 4.15) represents the kind of tourism objects that are present in the environment in which the visitor is to be found. It has been developed from scratch but taking into account the cDOTT ontology[BFP+09] and considering the different types of tourism services, i.e. accommodation, restaurants, etc. The ontology is further subdivided in the following concepts:

    - Tourism accommodation services
    - Tourism restaurant services
        - Covered restaurants
        - Fast food restaurants
        - Modern restaurants
        - Traditional restaurants
        - Vegetarian restaurants
    - Tourism transportation services

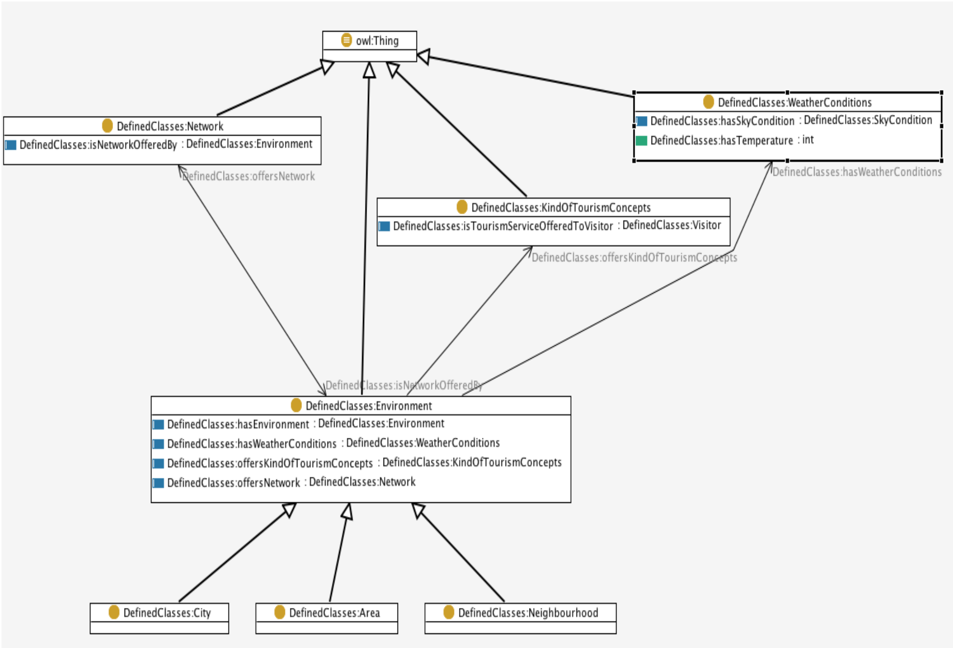

Figure 4.14: The environment ontology and its relations with other ontologies

This ontology has been linked, as shown in the previous bullet, to the *Environment* ontology. It has been assessed against CQs 57 - 65.

10. *Location*: This ontology (figure 4.16) represents the coordinates of the location of visitors at a given moment of time. It has been defined such as in CoDaMoS[PVR+06]. This ontology has been validated against questions 37 - 44.

11. *Weather Conditions*: This ontology (figure 4.17) represents the meteorological conditions at a particular location. It has been taken from the CoDaMoS ontology[PVR+06] and it considers the following concepts:

    - Sky Condition
    - Temperature

    The *WeatherConditions* ontology is linked to the ContOlogy network of ontologies through the *hasWeatherConditions*. This is an

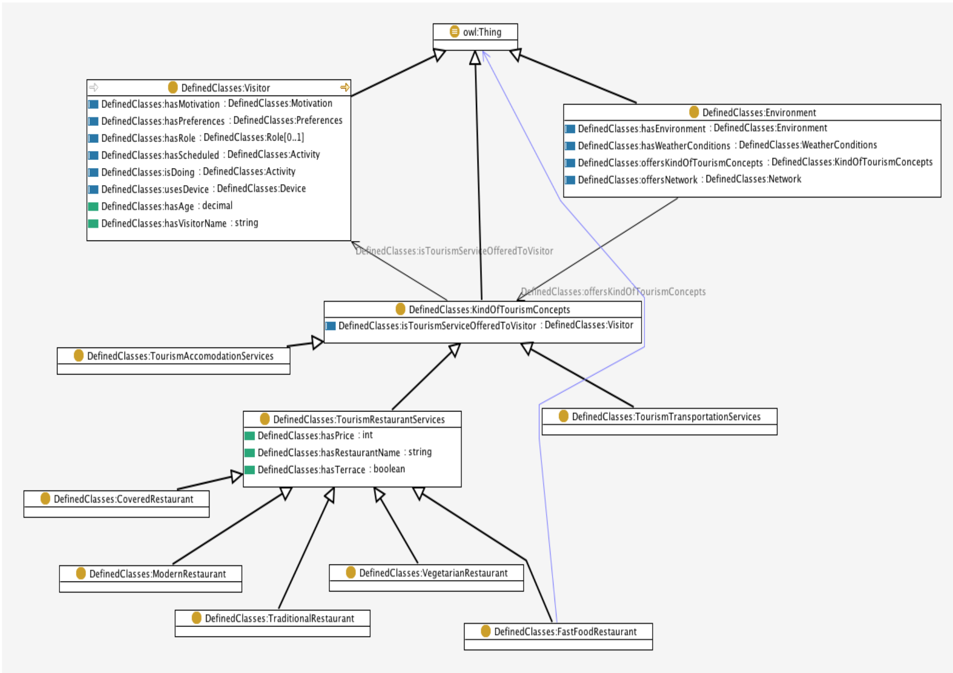

Figure 4.15: The tourism objects ontology and its relations with other ontologies

object property, which domain is *Environment* ontology and range *WeatherConditions*. Its structure can be seen in figure 4.17 and its validity has been assessed with CQs 51 - 56.

12. *Time*: This ontology (figure 4.18) represents the span of time between facts and it has been directly imported from the OWL-Time ontology[HP06] and incorporated on to the ContOlogy network of ontologies. This ontology has been validated with the CQ 45 - 50.

    A *hasTemporalEntity* object property has been added to the network of ontologies with free domain and range of *TemporalEntity*. This way, not only an *Activity*, but also an *Environment* or a *Task* can be framed in a time scale.

4.8 Summary

Ontology development is not an easy challenge. Both experience and literature show that not only are there multiple methodologies for building

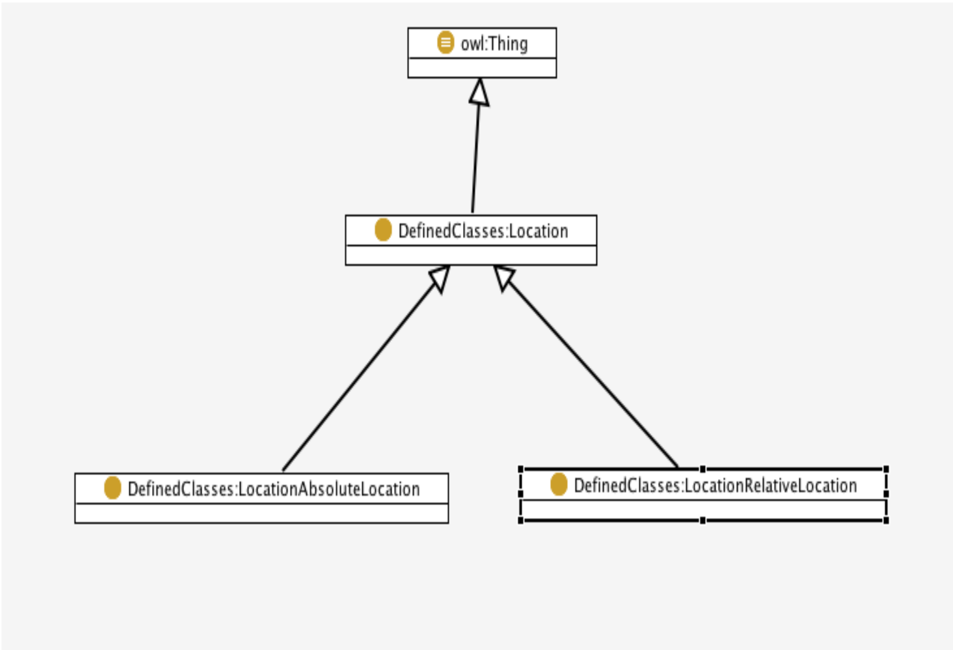

Figure 4.16: The location ontology and its relations with other ontologies

ontologies, but also that more than one ontology can be used for the same purpose. An example of this is the number of various ontologies that exist for context modelling and context information management[CFJ05][GPZ94][WZGP04][PVR+06][SRA06].
However, although they aim at representing something similar, i.e. context, they show the following drawbacks: Firstly, they are importantly influenced and bound to the particular requirements and constraints of the (portion of) reality they represent, which actually makes them different one from another but not automatically suitable for any domain. Secondly they are in early experimental phases, which does not provide them stability outside their "context" of use. Therefore, they are not automatically implementable in the realm of tourism as a whole, but only partly re-usable.

The importance of re-using already existing knowledge in scientific research is crucial. That is one the main reasons why existing ontologies (both context and non -directly- context related) have been re-used as much as possible, following the principles underlying the nature of ontologies[NFF+91] and highlighted in Section 2: *"ontologies allow to re-use and share common*

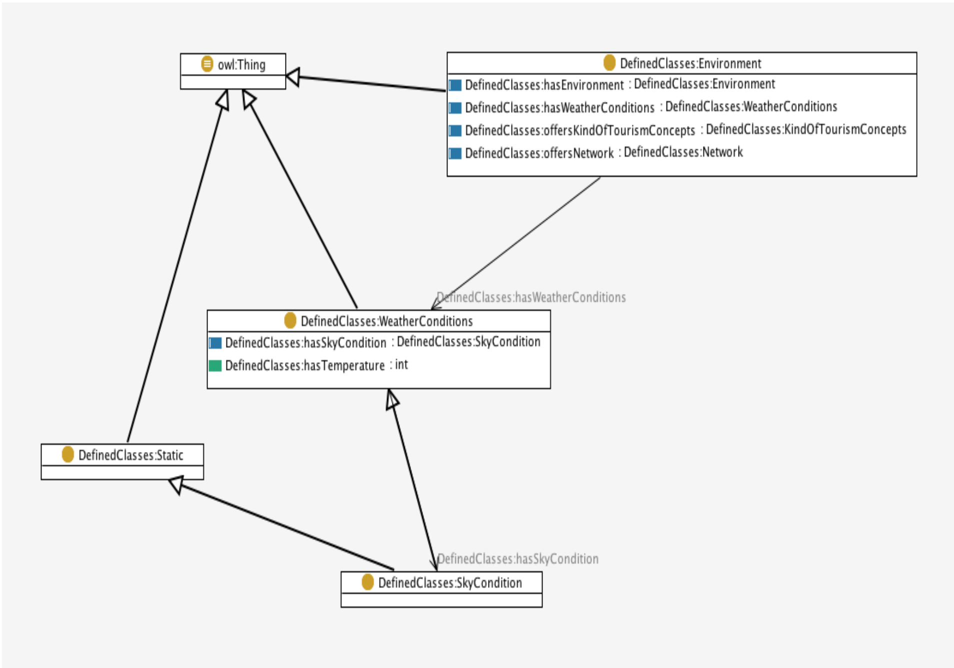

Figure 4.17: The weather conditions ontology and its relations with other ontologies

knowledge across applications, so that systems developers only need to worry about creating the specialized piece of knowledge and reasoners for the specific task of their system. This way, new systems did not have to be built from scratch, but by assembling re-usable components, saving this way time and money. System developers would then only need to worry about creating the specialized knowledge. Thus, declarative knowledge, problem-solving techniques and reasoning services could all be shared among systems".

In fact, the development of ContOlogy has primarily focused on appropriately addressing the information requirements established for human mobility in Section 3.4 and implementing them in the ContOlogy network of ontologies. In order to do this, some ontologies have been partially re-used, as indicated in Sections 4.6 and 4.7. And in addition, it has speeded up the ontology building process.

Ontologies have traditionally been used in pervasive computing for context modelling with five major objectives: (i) context knowledge re-use, (ii) interoperability, (iii) integration, (iv) sharing (machine understandability) and (v) reasoning to check the model's consistency and

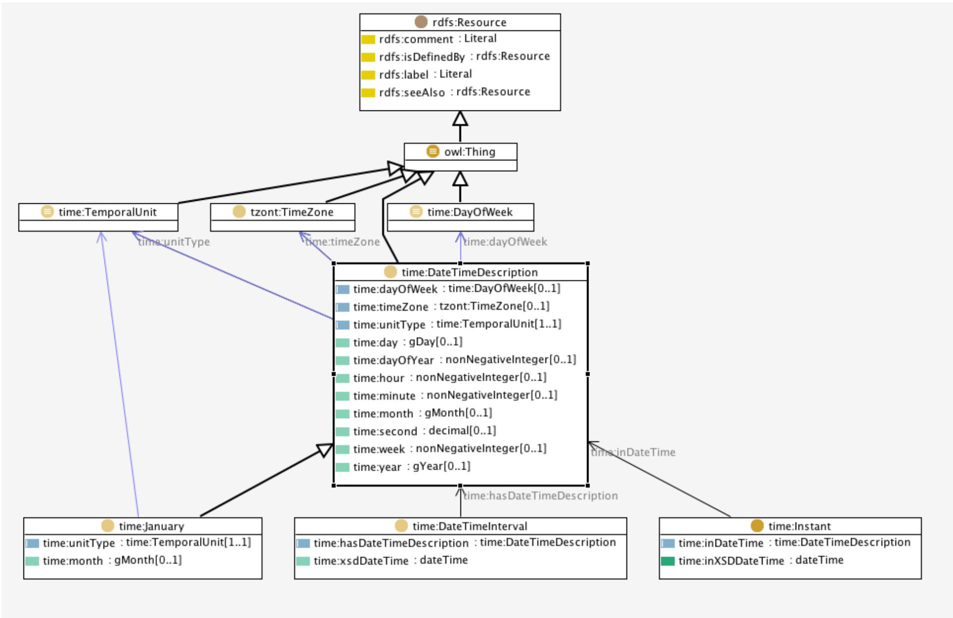

Figure 4.18: The time ontology and its relations with other ontologies

to infer implicit context knowledge. Ontologies provide interoperability at the model level as explained in Chapter 3. However, this interoperability is limited by the structure of the ontologies themselves, since their structure is dependant on the application in which they run and are not flexible enough to incorporate new knowledge, e.g. from a new ontology. Thus, aiming at reaching a higher degree of interoperability, it is not just necessary to use (single) ontologies, but the morphology of ontologies themselves needs to be more modular, flexible and scalable to really reach the desired level of interoperability at the model level.

In addition, based on the definition of context provided in Chapter 3 (Section 3.1.2) and to its further subdivision in two categories (A, B) the approach to a context ontology in the framework of human mobility cannot be represented by one single ontology. The context of a visitor is the aggregation and amalgamation of a number of constituents that have their own identity and entity as to be treated on their own independent way. Thus, a new ontology structure needs to be found. Each of these constituents can be defined and modelled by the use of ontologies and thus, an open, scalable context modelling methodology based upon a network of ontologies is based, rather than one single ontology or a double ontology (one core, one domain specific) approach[GPZ94][CFJ04b][WZGP04].The resulting ontology would have been too large in order to represent the information required and that would decrease its flexibility, modularity and scalability.

Various ontology building methodologies are roughly presented at the beginning of this Chapter, in Section 4.1. However, given the fact that (i) the conception of the notion of context leads to a modular representation of the model, (ii) that there are ontologies that can be (partly) re-used in the process and that (iii) a higher degree of interoperability is needed at the model (knowledge) level has totally determined the ontology building methodology to use, i.e. the NeOn methodology for building networks of ontologies. Moreover, not only does NeOn provide guidelines to build modular ontologies (in form of networks) and to re-use existing ones, but in addition it sticks to the standard software development requirements provided by the IEEE.

The NeOn methodology has been briefly presented in Section 4.2 and then, its deployment in the building process of the ContOlogy network of ontologies has been adapted as shown Section by Section (Section 4.5.1 through Section 4.5.7). Since the ontology building process is iterative, following the NeOn methodology guidelines the ContOlogy network of ontologies shall be implemented in three consecutive iterations. Each of them will provide a working prototype of the network of ontologies suitable for model validation purposes.

Section 4.6 presents the result of the previous process. It shows the ontologies that comprehend the ContOlogy network of ontologies as well as a high level graphical overview of its structure. Finally, Section 4.7 shows the validation of the network of ontologies. Indeed, the network of ontologies is valid according to the NeOn evaluation methodology, since the components of the ontology answer the competency questions proposed in the requirement specification Section.

Chapter 5

CONCERT Framework Architecture

"You see things, and you say: 'Why?' But I dream things that never were, and I say 'Why not?'"
George Bernard Shaw, (1856 - 1950), Irish playwright. Awarded a Nobel Price for literature in 1925 and an Oscar in 1938.

Building contextual computing systems and applications involves several challenges, such as gathering, disseminating, modelling, storing and managing contextual information. All of these challenges and issues justify the need for an architectural support to provide an efficient infrastructure, not only for building these kind of systems, but also, to be able to replicate them outside their original conception environment.

There are entire works that have been dedicated to study context-aware architectures [SAW94] [DA00a] [CK03] [CFJ04b] [Bal07] [Bat07] [Euz95] [KPM05] [Pas97] [BMU02] [PS07] [RSC07] [Rib06] [SNLL05] [dSSRC+07] [VP00] [VSS+07]. Although various initiatives have focused their research on trying to separate context gathering from context management, still these systems are dependent on the particular infrastructure needed to gather contextual information, i.e. sensors, their hardware infrastructure and corresponding communication protocols. As well as it happens with existing context ontologies and shown in previous Chapters, these dependency issues have proven to make context-aware systems neither as stable nor scalable as desired.

In the case of the CONCERT contextual computing framework, the architectural support is not only important because it schematically shows the components that configure it as well as the interactions among them in terms of relations, but because it is an independent architectural layer with respect to the context information gathering and dissemination infrastructure required in any context-aware system. This way, the CONCERT architecture is based on a layered distribution (see figure 5.1) in order to separate low-level tasks, i.e. discovering, gathering and dissemination of context information to the context information processing module (context manager) from high level tasks (effectively managing context, querying, reasoning and consistency check).

As stated earlier, both context and tourism information are gathered from the Internet, i.e. anywhere and anytime, hence not restricting the use of contextual computing systems to sentient environments (Chapter 3). Once gathered, the information needs to be transmitted to the context management module, for it to process and finally deliver the relevant piece of information to tourists. All of these architectural details are thoroughly described in Section 5.1.

Therefore, the transmission of context and tourism information to mobile devices has to be wireless, for no hardwire infrastructure can be established between the sources of contextual information and the context manager, which on the case of CONCERT resides in the client side (Section 5.2.2). Traditional communication technologies such as 3G, GPRS and UMTS could be used for dissemination purposes, however, they have two important drawbacks: First, they are typical *pull connection* technologies, i.e. they have not been thought to automatically and continuously send information, but for users to actively be connected to sources of information and retrieve the required piece of their interest. Second, as stated in the introductory part of Chapter 1, their use is very expensive unless a flat rate is arranged with the corresponding telecom operator. Their use is even more expensive under roaming conditions, where prices can go up to 1 Euro per MB. Another issue that needs to be considered is that of coverage, however, it is beyond the scope of this dissertation.

This is one of the reasons why the CONCERT framework experiments with another communication technology, i.e. digital broadcasting. This technology is still in early experimental phase for data transfer, however, as mentioned in Section 1.3, it is an efficient *push* technology highly appropriate for the realm of tourism[HFZ+06][HSLF08][BFH+07]. One drawback of using digital broadcasting is that it does not have an up-link communication channel. Thus, visitors would only be able to receive information by these means, but not sending it back to a source of information in form of queries, requests and so on. Another kind of communication technology, such as 3G, Wi-Fi, would be needed for this purpose. However, how digital broadcasting can work simultaneously in combination with other communication technologies is something on which intensive research efforts have to be invested, as will be explained in Section 7.3. Hence, Section 5.2 explains the features of a typical broadcasting architecture as will be used in the framework of this dissertation. It is important to remark at this moment that digital broadcasting shall be used as a tool, not as a research subject. Therefore, although some technical details shall be provided throughout this Chapter on DRM in particular, it is far beyond the scope of this dissertation to further discuss or deeply analyse on technical matters of digital broadcasting. Thus, the Section briefly reviews DRM standard systems, the DRM Content Server and thoroughly describes the structure content needs to have (XML and JML, detailed in Sections 5.2.4 and 5.2.5 respectively) in order to be broadcasted via MDI (Multimedia Distribution Interface) protocol.

Section 5.3 describes an architecture that results from the combination of the original CONCERT contextual computing architecture (Section 5.1) and the traditional Digital Broadcasting Architecture depicted in Section 5.2. It shows conceptually how broadcasted data can be manipulated, i.e. *filtered*, by the CONCERT architecture to provide visitors with context-based information. This Section is particularly relevant, since it provides, at least conceptually, the answer to one of the research questions suggested in Section 1.4, namely whether semantic technologies can be used to filter incoming broadcasted information.

Based on Section 5.3, Section 5.4 explains the CONCERT based application and its different components. This application has been built in order to test and assess the theoretic framework presented in Chapter 3. The first component to be described is the RSS Maker, an interface that allows to generate tourism content and uploads it to the Content Server. Following, the filtering engine is explained, i.e. how the ContOlogy network of ontologies effectively filters the information by means of a set of rules defined on top of it. The visualization interface, i.e. CONCERT viewer is shown next. And finally, the workflow of the application is shown through some image captures. This interface is an emulation of what visitors would see in the screen of their mobile devices and displays the tourism information they would receive depending on their context. It also shows roughly how visitors' interaction with it would be.

The Chapter finishes with a summary in Section 5.5, where the corroboration of the first two hypotheses, as well as the answer to the research questions can be found.

5.1 CONCERT architecture

The CONCERT contextual computing architecture (Fig.5.1) is based on a layered distribution in order to separate different important tasks involved in any context-based application, i.e. context gathering by context providers and context processing modules.

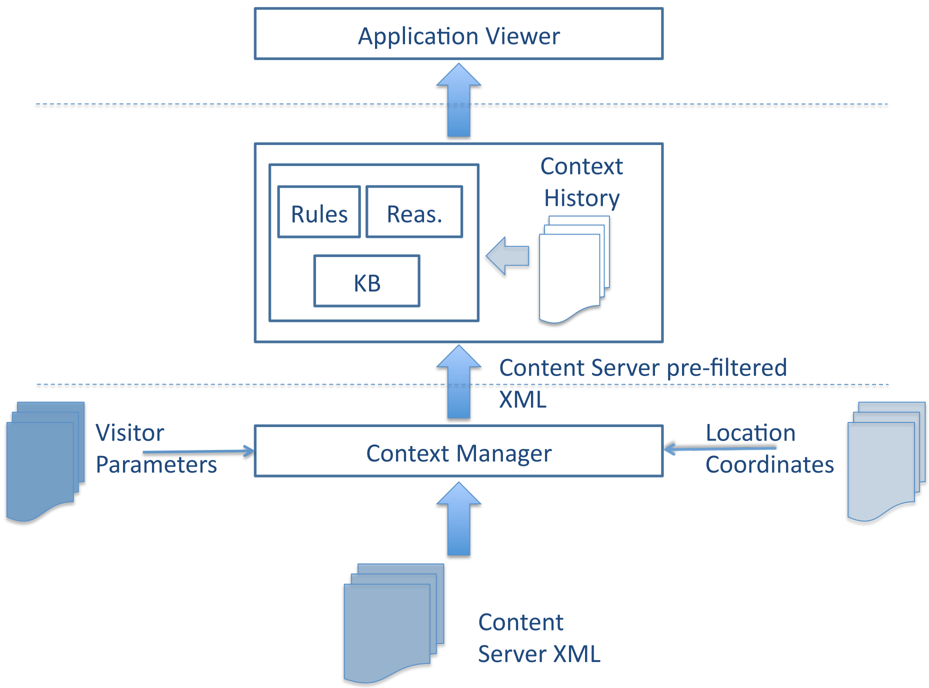

Figure 5.1: CONCERT Contextual Computing architecture

The different modules and components in the architecture are interconnected as shown in figure 5.1. Its constitution has been highly influenced by the theoretic conception of the CONCERT framework, described in Chapter 3. The three theoretic contributions described in that Chapter have direct implications in the CONCERT architecture as follows:

- *Definition of context*: The new theoretic approach to context explained earlier, led to a new definition of the notion of context[LASM+09a], which at the same time, determines the input parameters to the context manager. These input parameters are visitor parameters, general context and tourism information from the Internet (specified in this particular case but not subject to the Content Server XML file, Section 5.1.4), and location coordinates or other potential sources of information directly provided by user devices.

- *Interoperability at the infrastructure level*: The fact that no sensors are used allows to use a less complex architecture in terms of hardware, since all incoming information from the Internet is reduced to one XML file.

- *Digital broadcasting*: The fact that the CONCERT framework experiments with digital broadcasting implies that the mobile device used by visitors must be laptops, since no DRM receiver has been yet made available for mobile phones. This means that the CONCERT architecture will be hosted in the client, and allows the context network of ontology to be larger than it would if it had to be hosted on a real mobile phone type of device. In addition, the interface provided by the CONCERT Viewer (figure 5.15) allows users to intuitively introduce their personal parameters. These personal data shall be hosted in the client.

The context model used in CONCERT does not have any effect in the architecture, since it is located in a middle layer and is not part neither of the input nor of the output of the architecture. In addition, it is important to remark that this architecture is not only independent from the context gathering infrastructure, but also from the particular sources of context information, as they reside on the Internet. All context and tourism information constitute an input parameter to the context manager and contribute to provide the CONCERT architecture independence from any fixed infrastructure.

The components of the contextual computing module shown in Figure 5.1, are explained as follows (some of the components have been omitted from the figure for clarity):

- *Context Manager*: This is the central component of the Contextual-Computing layer. This software module is a background process that runs on mobile devices (client). The Context Manager receives information from the Context Providers (information input), such as the location sensor embedded in the mobile device, the wireless broadcast source and the user preferences, provided manually by the user. The Context Manager pre-processes all of the gathered information and consequently feeds the Knowledge Base with that information triggering the flow of events, which are necessary to provide visitors with relevant information for their needs. It offers a centralised way to access context data sources. In addition to this, the Context Manager also runs another service, which consists of removing every certain time all of the incoming information in order to avoid information overloads.

- *Context Providers*: Context Providers are software adapter modules, which specialize in a specific type of context information. They capture the information and send it to the Context Manager defined earlier. There are three context providers in the CONCERT architecture, from which one of them can be treated as a multiple context source, since it provides information from various distinct information sources in Internet:
  - *Location information*: The Location hardware present in the mobile device, feeds the system with visitor location information. It provides the information in typical GPS coordinates format. This information is then transformed into "logic location" and added into the knowledge base this way. It is important to remark that in case of GPS failure, the location can be introduced manually (see figure 5.15, CONCERT Viewer).
  - *Visitor Information*: Another context provider is in charge of maintaining an up-to-date list of user preferences, that visitors are expected to provide to the system via the CONCERT viewer, figure 5.15. The application layer features an interface (Figure 5.15) where users can update their personal parameters such as preferences, motivations, etc. to enhance the information filtering process. It is worth mentioning again that the visitor information corresponds with the guidelines provided by the UNWTO.
  - *Sources of information*: Finally, a third context Provider handles the reception of Internet based context and tourism information. In the case of the CONCERT framework, The Journaline XML file received by means of digital broadcasting provides this context information. External entities are used to acquire context data from heterogeneous sources, e.g. Web sources, such as weather Web services, or from the traveller's mobile device, e.g. profile, location. This XML information, together with the other context information are fed into the context manager for its pre-processing and aggregation in the knowledge base. This is one of the novelties within the CONCERT framework: the use of potential CONCERT-based application is not going to be limited to a

particular predetermined sensor-populated location. On the contrary, it could be used anywhere where there is telephone network coverage (in this particular case, digital broadcasting coverage) that enables access to information sources.

- *Knowledge base*: The knowledge base is the core part of the CONCERT architecture. The Knowledge Base configures the second layer of the CONCERT framework and it contains the following components:
  - *ContOlogy*: This is the network of ontologies that comprises the ontologies themselves and instance data, which models visitors' context, as specified in Chapter 4. Initially, the knowledge base is populated with static information regarding the environment and the services offered in the environment where the application is running. Gradually, as the dynamic flow of context information reaches the context manager, it is sent to the knowledge base after pre-processing. The context manager checks whether a particular class of an ontology within the network has already been instantiated. If it is the case, it then checks the value of the instance and compares it to the new one. If they are equal, the context manager does not perform any action, else it first removes the existing variable, sends it to the context history database and then inserts the new one.
  - *Rules*: The rules have been implemented on top of the network of ontologies in order to filter tourism information according to the particular value of context at a given moment of time. They are used by the reasoning/inference engine presented in Section 5.4.3.
  - *Inference engine*: It is used to obtain high level context (situations) based on defined rules and the semantics of information that has been gathered and stored in the Knowledge Base. As dynamic context information reaches the network of ontologies, the reasoning engine filters tourism information based upon the rules that govern its behaviour (Section 6.2). It uses the specific values of context constituents at that moment in order the filtering to match users' requirements at that same moment of time. It stores all the statements about tourist's context by the use of ontologies.
  - *Context History*: The context history repository contains data about past values of context. This information can be of use to support and enhance the filtering process. The context history

can also be useful to predict future traveller situations by the use of the Context History Exploitation Engine or to reason over current values of context variables. One possible application of use can be the following: for example, if the coordinates given by the mobile device GPS incorporated sensors correspond to Athens and they do not exist in the Context History database, therefore, the knowledge base can further infer that the traveller is in Athens for the first time.

  – *Access manager*: It manages the interaction between the platform and the application layer. This interaction can be in a request/response manner or in subscription basis, where the platform sends context information according to defined events (context changes, time intervals).
  – *Query engine*: It allows queries about context information, as location, temperature or higher level context.
  – *Privacy, Trust and Security Control*: Given that contextual information may contain very sensible private information, a security and privacy module needs to be implemented within the architecture. This way, travellers themselves are the owners of the information and can decide the extent to which they want to share personal information either with others or with the system. Past context-aware initiatives have also worked on data privacy[CFJ04b][DA00a].

- *Application Module*: The third level of the context architecture is composed by the application layer, which consists of the presentation logic that interfaces between the contextual computing system and the user. The application features an embedded Web browser where the result of the filtering (an output HTML file containing only context-based information) is displayed as shown in figure 5.15, CONCERT Viewer.

5.2 Digital Broadcasting Standards

As mentioned both in the introduction of this dissertation and at the beginning of this Chapter, digital broadcasting has been chosen as the communication technology to disseminate both context and tourism information from their sources to mobile devices. There are three standards in this realm, namely DAB, DRM and DRM +.

DAB is the first terrestrial digital radio broadcasting standard system and is intensively used in Europe. It was designed during the 1980s in Germany and is considered to be the digital equivalent to FM (Frequency Modulation), well known for its high quality sound transmission. However, due some commercial problems, this standard has not yet been widely implemented. Nonetheless, there are some signs of recovery. For example, in the case of UK the segmentation of different contents and the inclusion of data-based services complementary to audio services, have produced a 20% increase of the DAB market share. In Germany, the intensive adoption of information-based systems by the strategic automotive sector in terms of real-time traffic information, multi-channel audio and embedded receptors in (very) high-range automobiles, has mobilised a number of agents within the value chain, from chip manufacturers to both private and public broadcasters. As a consequence, and given the official support provided by governmental institutions and regulations, the coverage of DAB has increased up to 90% of the territory.

On the other hand, DRM is intended to complement AM (Amplitude Modulation). AM was implemented at the beginning of the 20th century. The number of broadcasters and users suffered a dramatic increase, and as a consequence there are over 2 billion AM users worldwide nowadays. Thus, most of the regions in the world have access to at least basic AM radio services. In addition, these services are not only received from the country of their origin and therefore, there are an important number of programmes in long-, medium-, short-wave bands under AM frequencies.

Finally there is the DRM+ standard which can be considered an extension of DRM, since it is based and supports all DRM services and characteristics. The main difference lays in the fact that it works on frequencies up to 174 MHz, occupying spectral bands up to 100 kHz, whereas DRM works on frequencies up to 9 or 10 kHz and in a spectral band of up to 30 MHz. Summing, this is translated in a lower coverage range than DRM but a wider bandwidth up to 100 kbps.

In the framework of this dissertation, the Digital Radio Mondiale (DRM) terrestrial digital broadcasting standard has been chosen as experimental tourism and context information distribution channel. In addition to that, a prototype based on DRM has been developed to show the potential this communication standard has in order to meet the needs and requirements of the tourism sector, and in particular of the CONCERT framework, providing a differentiate value in various ways.

DRM uses frequency bands up to 30MHz. The frequency bands used for broadcasting below 30 MHz offer portable and mobile reception and large coverage areas - especially high frequency (HF) bands, which offer the possibility of international broadcasting without using local repeater stations (contrary to mobile telecommunication technologies). However, broadcasting services in these bands, which use analogue techniques, have a low audio quality and are subject to considerable interference.

The DRM consortium[1] was founded in 1998. It is an international non-profit organization composed of broadcasters, network providers, transmitter and receiver manufacturers as well as universities, research centres and other kinds of organizations. The DRM system was first standardized in 2001. DRM's main goal is to set a worldwide standard for digital radio in long-, medium- and short- wave bands, in order to increase the reception quality in all bands below 30 MHz.

The International Telecommunication Union (ITU)[2] recommends DRM digital transmission system, since it does not only offer audio broadcasting in near-FM quality, but also different types of multimedia data services, which are particularly interesting for the realm of information-based tourism services, where information can be sent in form of pictures, videos or podcasts, for example. The system can be applied to all AM broadcast bands below 30 MHz (LW, MW and SW) and it is understood as a one-to-many delivery system, allowing to increase the number of concurrent users, ideal for use by people on the move and not subject to real connection technologies, dependant on server infrastructures, potentially increasing the risk of failure, that minimizes the amount of concurrent users.

DRM meets all stringent requirements laid down by the ITU for digital broadcasting systems, as included in ITU Recommendations BS 1514[3]. It is also an agreed European Standard (ETSI ES 201 980) and is adopted by the European Telecommunication Standards Institute[ETS08].

DRM is intended to substitute the current analogue AM transmission systems, thanks to the improvements its use involves. While it is true that AM allows simple receivers and has an average of 2.2 billion receivers worldwide, it also faces a number of challenges. Its poor audio quality (due to a small bandwidth), its out-dated receiver handling (no station label, no automatic frequency switching) and its high power consumption for transmission are just some of the important issues the DRM consortium is currently working on. FM has a good audio quality compared to AM, but a far smaller coverage area.

[1] http://www.drm.org

[2] http://www.etsi.org

[3] http://www.itu.int/md/R00-WP6E-C-0126/es

	f	power	coverage	data transmission
DAB	>30 MHz.	50 kW	Low	<350 kbps
DRM	20 +/−10 MHz.	50 kW	Medium	48-180 kbps
DRM +	30 - 174 MHz.	50 kW	Large	40 - 186 kbps

Table 5.1: DAB vs. DRM

Compared to conventional analogue systems, DRM offers very important benefits for the users. Some of them are detailed here:

- Interference free reception on short, medium and long wave.
- High audio quality ('full' audio frequency spectrum; comparable to FM quality).
- Sender label, text messages, data services, multimedia.
- Automatic frequency switching simple user interface.
- One receiver for worldwide operation Large coverage areas - e.g. reception of home radio stations on vacation.
- Inexpensive receivers available in the near future.

In addition, broadcasters (who in the case of this dissertation would be tourism boards and commissions and Destination Management Organizations (DMOs) can also largely benefit from this standard in the following way:

- Support of ground wave, sky wave and NVIS (Near Vertical Incidence Skywave).
- Bandwidth support for 4.5, 5, 9, 10, 18 and 20 kHz transmission channels.
- Support for single frequency networks.
- Up to four programmes/services on one frequency.
- Consumed transmission power only 1/4 of analogue AM.

5.2.1 Justification of use of DRM

Considering what has been said about digital broadcasting so far and bearing in mind the CONCERT's main objectives, the following arguments are provided to justify the election of this standard amongst the various existent ones:

- DRM is an open and free standard and does not entail royalty or licensing fees for its use to users and/or receptor manufacturers. Nonetheless, there may be other licences or royalties related with coding systems, or patents on DRM-based systems that need to be paid in case of use.

- DRM systems operate on a frequency band up to 30MHz, which makes this technology ideal in order to broadcast content in large areas in relation to the power and energy needed for these transmissions. This would be a good characteristic to take advantage of in the realm of tourism, since broadcasted information could be received in any country without concern on roaming prices.

- Although the following is a general characteristic of broadcasted services in comparison to other connection-based (*pull*) communication systems, the robustness of the DRM channel is especially remarkable in low and medium wave transmissions. In fact, in critical situations, such as natural disasters or events of war, short and medium wave band radio have been used as communication channels towards the people. These characteristics make out of DRM an ideal standard in order to broadcast not only tourism related information, but any other critical information independently from complicated infrastructures. It is important to bear in mind, that the tourism sector is especially sensitive to all information related to security, especially in particular destinations in the World, therefore, this feature of DRM is not trivial.

- Another key characteristic of DRM as ideal communication system in the realm of tourism is its competitive positioning from the access and economic points of view, against other existing communication systems nowadays. The CONCERT based application developed in the framework of this dissertation covers the gap left in terms of coverage by cellular technology or VHF on the one hand, and the gap left by satellital DRM due to high cost of use reasons.

	Communication Technologies	Digital Broadcasting DRM
Price	High - up 10 EUR 1 per Mb	Low - free[4]
Coverage	Country - shades	Various countries[5]
Data Transfer	High	Low
Infrastructure	High	Low
Nature	Pull	Push

Table 5.2: Comparison: communication technologies vs. DRM

5.2.2 Overview of DRM systems

For the broadcast signal, the DRM system uses a high number of QAM (Quatrature amplitude modulation)[ETS05c] modulated carriers spread over a regular AM (SW, MW, LW) spectrum channel. In order to generate the signal in the air on the transmitter side, the DRM transmission block diagram consists roughly of the stages explained next:

- First of all, on the source encoder and data pre-coder, the signal is adapted to an appropriate digital format (Section 5.4.2). This source data stream is then combined in the multiplexer with descriptive information about the RF-signal (radio frequency signal), the transported data and services and additional functions. The multiplex can support up to four audio/data streams.

- The next step is the channel encoding, which adds redundant information in order to permit the receiver to reconstruct a distorted signal and defines the mapping of the digital encoded information (the DRM multiplex which is provided by the DRM Content Server) onto QAM cells. After that, a set of QAM symbols is interleaved, so that adjacent QAM cells are spread, before the transmission, across the carriers and re-ordered again in the receiver in order to avoid possible long blocks of lost data. Thus, the robustness of the bit stream to channel errors is highly improved. Then, in the OFDM (Orthogonal Frequency Division Multiplexing)[ETS05c] cell mapper, the different types of QAM cells are collected and placed on the time/frequency grid. OFDM symbols are separated by guard intervals between every symbol.

- And finally, the modulator converts the digital representation of the OFDM signal into the analogue base band signal.

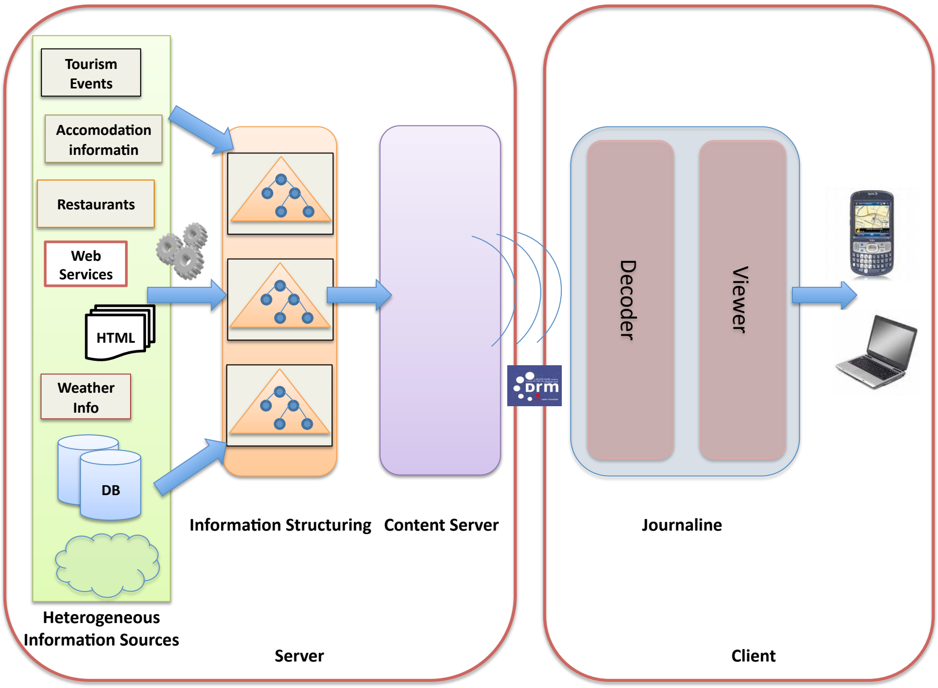

Figure 5.2: Typical Digital Broadcasting architecture

Based on the previous, figure 5.2 shows the overview of a typical broadcasting service architecture adapted for the particular case of the CONCERT framework. The server side gathers information from heterogeneous distributed sources in Internet. As stated earlier in Section 5.1. This information may contain data about tourism sightseeing offers, restaurants, accommodation, agenda, museums, as well as other context information, such as weather information, etc. These sources can be simple (unstructured) Websites and/or XML files[6], or official structured destination management organizations' (DMO) data specifically uploaded for public consumption.

Once this information has been gathered, it has to be structured in XML file that follows a specific schema, Section 5.2.4. This schema has to be compliant with the defined Journaline schema[ETS08], so that it can be uploaded to the content server[FI08]. The Content Server compresses

[6] http://www.w3.org/XML/

and reformats this file into a JML (Journaline Markup Language)[ETS08], Section 5.2.5, and finally broadcasts it via MDI (Multiplex Distribution Interface)[ETS05b]. This has to be done in this way, since broadcasting bandwidth is often limited by the configuration of the Content Server's carousel.

The Journaline module, which is located on the client side (in the case of the CONCERT framework, on the visitors' mobile device) receives the information that is being continuously broadcasted by the Content Server. Journaline implements two functionalities:

- First, it decompresses the information that has been received in JML format back to XML in the decoder module.
- Then, the resulting XML file is further transformed into HTML and the file would then be ready to be browsed in a typical Web navigator located in visitors' mobile devices.

Journaline is the standard developed by the Fraunhofer Institute for Integrated Circuits[7] (Fraunhofer IIS) for data transmission for DRM broadcasting channels, as well as for other digital broadcasting systems such as DAB/DMB[ETS08]. The use of Journaline guarantees the optimized use of the radio channel for information distribution through the JML (Journaline Markup Language) (section 5.2.5), a simplified version of the XML. It also assures service compatibility with DAB/DMB in all kinds of mobile devices such as Smartphones, PCs, music players, PDAs enabled with DRM/DAB/DMB receivers.

5.2.3 DRM Content Server

After having briefly described a general digital broadcasting architecture, this Section presents the Content Server and shows its features without getting too much into detail. The DRM ContentServerTM R4 is a highly reliable professional broadcast system for Digital Radio Mondiale[ETS05c]. It provides all options DRM offers and all interfaces for a smooth integration into the broadcast chain. The DRM ContentServerTM R4 provides the following functionalities:

- DRM AudioServer with multi stream real-time audio encoding.

[7] http://www.iis.fraunhofer.de/index.jsp

- DRM Multimedia DataServer providing all standardized as well as proprietary mechanisms for data collection, import, processing, encoding and broadcast.

- DRM Multiplex Generator managing the extensive DRM signalling capabilities, generating the full digital DRM Multiplex and providing standard MDI/DCP (Distribution Communication Protocol) output streams.

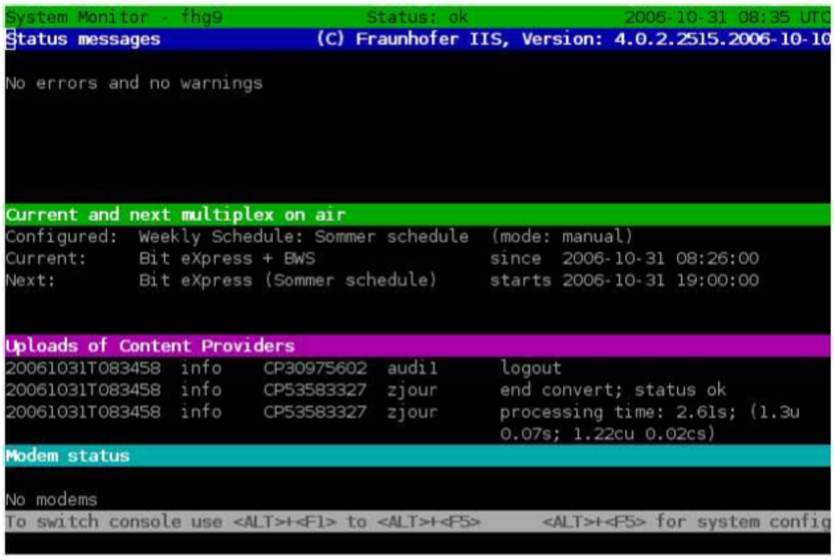

Figure 5.3: Screenshot of the Content Server

For the purposes of the experiments carried out in the framework of this dissertation, a Content Server has been installed in a PC in a laboratory environment. A screen shot of the Content Server can be seen in figure 5.3. The Content Server behaves as an autonomous operating system running on a Linux machine. It configures 5 consoles as follows:

- A system monitor showing the status of the system.

- A text based Web browser to locally browse the Content Server documentation.

- Three standard Linux consoles to log into the Content Server or configure it.

The Content Server is typically located at the transmitter site, with full remote control for administration and data provision. In potential use within the tourism sector, it would be installed in any official radio sender, Destination Management Organization (DMO) or any other official institution for purposes of tourism content provision. The Content Server has been implemented with a functionality that features a remote Web interface, so that it can be accessed through and from any modern Web browser. The output signal of the DRM ContentServerTM R4 carries the complete DRM Multiplex (FAC, SDC, MSC) in MDI/DCP format according to ETSI TS 102 820 (Multiplex Distribution Interface)[ETS05b] and ETSI TS 102 821 (Distribution and Communications Protocol)[ETS05a]. This DRM Multiplex can be fed simultaneously to any number of DRM Modulators (optional SFN operation) and monitoring stations.

5.2.4 Journaline Content Structure

Journaline is a text based information service for digital radio, optimized for simple data aggregation and re-use, as well as for highly efficient broadcast transmission. It supports the widest range of receiver types, from low-cost solutions with a small text display up to high-end receivers with graphical user interfaces and optional text-to-speech playback. The information provided by the Content Server and received by the Journaline module can also be displayed in a typical Web browser.

The radio user can instantly and interactively access all information provided by the radio station, comparable to teletext for TV, in a simple navigation schema (see figure 5.4) . The core information is provided in simple textual form with the option for richer graphical representation including a future extension to multimedia elements like images or video sequences. The flexible coding of Journaline allows for future extensions as well as the provision of additional non-textual information.

The information is hierarchically organized based on menus. Every menu contains a list of sub-menus and/or messages. Messages carry one piece of information each (e.g. a list of restaurants, a news item, an event taking place in the evening, a traffic message, and so forth.). An example of such messages is shown in figure 5.4.

Journaline is based on common public standards:

- Objects can be generated and fed into a Journaline enabled transmission system as XML files.

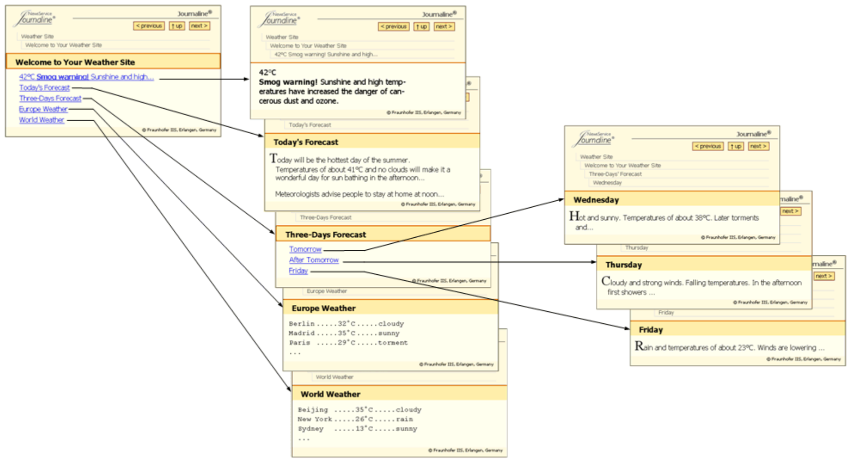

Figure 5.4: Example of Journaline information objects

- Journaline capable receivers can easily restore an XML representation of each object for further processing, or directly parse the transmitted binary representation for increased efficiency. In fact, this is the case of the CONCERT-based application that is presented in the following section.

Journaline is also highly optimized for use in digital broadcast systems and provides:

- *Added value for listeners*: Immediate access to textual information everywhere.

- *Ease of use*: Simple navigation through a minimum user interface consisting of a text screen, and up/down, select and back keys, similar to the functionality of any basic Web browser.

- *Great benefit for all types of receivers*: From low-cost receivers with small text display to high-end receivers with graphical user interface and optional speech playback. Also Web interfaces.

- *Increased listener loyalty for broadcasters*: Minimum additional bit rate for extended listener support.
- *Automatic re-use of existing information sources and feeds*: Many text-based information sources like RSS feeds can be broadcasted with minimum transformation effort.

The following technical principles ensure the stated functional highlights:

- All aspects of the Journaline specification are tailored towards low-footprint implementations even in low-complexity receivers as well as ease of use.
- To reduce the decoder footprint on low-complexity receivers, an object's main visual content is formatted as pure textual information.
- Every object is self-describing and self-contained. No global data structures need to be assembled or maintained in the receiver.
- Objects are broadcast in form of a data carousel. Data caching in the receiver is supported but not mandatory.
- Object transmission repetition can depend on individual access time requirements (priority class concept). For example, menu objects may be broadcast more frequently than message objects.
- To prevent unnecessary data overhead (and thereby minimize transmission time), objects are re-formatted for transmission into JML - the XML structured binary encoding "Journaline Markup Language".
- The receiver behaviour guidelines are a recommendation that should be fulfilled by any receiver according to its abilities and technical infrastructure. However, Journaline has a very carefully selected set of required functionality that encapsulates the core use to the user and that can easily be implemented by the simplest receiver.
- Receiver implementations can easily distinguish themselves by providing for example a richer graphical representation, Favourites functionality, or additional navigation support (like next and previous keys to speed up navigation through messages within the same menu).
- All aspects of the Journaline specification can easily be extended in a fully backward compatible way to support new types of information or extended functionality.

5.2.5 JML objects

Previous sections (Section 5.2.3 and Section 5.2.4) have introduced the notion of JML and the particular "Journaline" structure content needs to follow in order for it to be broadcasted by the Content Server. This Section will now define and describe JML objects, as well as their structure and main characteristics.

Each piece of information (i.e. every menu, news article, ticker message, etc.) is transmitted from the Content Server to a DRM receiver as a self-contained and self-describing "JML object", as mentioned in the previous Section. To minimize the decoder footprint in the receiver, the receiver does not need to build or store any overall hierarchy or index information to render a JML object on screen and to provide the user with the defined methods of interaction (like menu selections).

JML ("Journaline Markup Language"), an XML based binary coded content representation, structures JML objects' content into blocks of information like title, body text, menu items, etc. Journaline currently specifies 4 types of JML objects:

- Menus.
- Plain text messages.
- Title only messages.
- List messages.

Figure 5.5 shows the conceptual representation of JML objects, more graphically exemplified in figure 5.4. The logical structure of Journaline services is presented to the user as a tree of menus, sub-menus and messages. These JML object types differ in their preferred way of presentation to the user as well as their behaviour in the case of a content update being received while the JML object is displayed. The JML object types also support individual types of JML information blocks. For example, a menu consists of one title block and a range of menu items, while a title-only message only carries a title block.

Encoded JML objects of type menu are typically referred to as "menu objects", while the different types of messages are known as "message objects".

A JML object of type menu ("menu object") consists of a title block followed by a list of linked items. Linked items consist of a visible label plus a reference to another JML object.

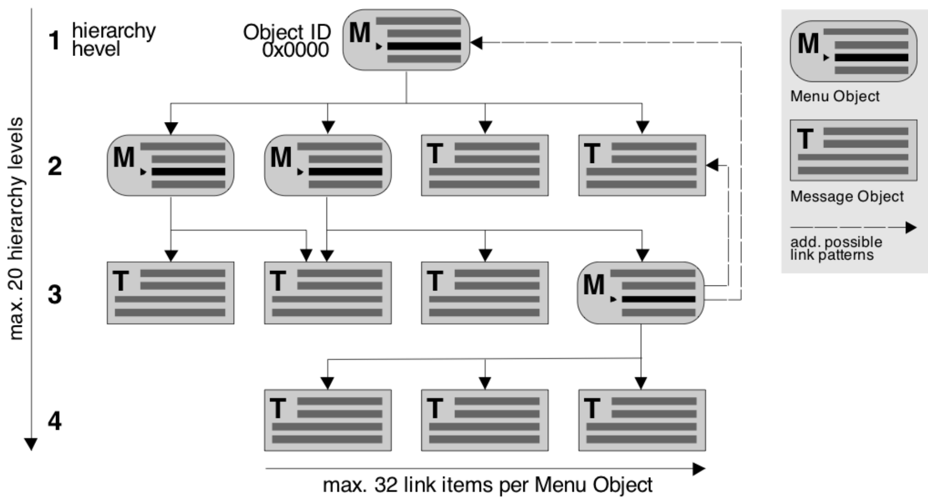

Figure 5.5: JML object hierarchy

The user must be able to navigate through the list of link items and select one of them. If the user activates one of the provided links, the referenced JML object shall be presented (thereby replacing the currently visible JML object).

Plain text messages consist of a title block followed by a block of body text. This type of JML object is intended for the presentation of longer messages.

When presented to the user, the title text and the following body text should be rendered on the screen with automatic line wrapping. The user must be able to access the full text (e.g. by manual scrolling). The title may be presented attached to the text (and therefore eventually scroll out of view along with the body text).

All JML objects that form a Journaline service can be distinguished by their unique object ID. A Journaline service must at least comprise one JML object with the object ID value 0x0000, the "root object".

When the user initially accesses a Journaline service and no point of entry is explicitly defined by other means, the root object is presented. Typically this root object will be a menu object (the “main menu”), but it could also be any other type of JML object.

Every JML object must be referenced from at least one other JML object, with the exception of the root object and JML objects that serve as alternative points of entry to the service (accessed by external means).

Every JML object may be referenced from multiple JML objects of different hierarchy levels (for example the main menu might be referenced from every menu object in the service).

5.3 Semantic-based contextual computing architecture

One of the two general research questions suggested at the beginning of this dissertation (Section 1.4) suggests whether digital broadcasting can be used in order to provide context-based information to people on the move. The previous Sections in this Chapter have described and analysed both the CONCERT architecture and the digital broadcasting architectures, as well as the requirements in terms of information structure. The issue now is how to combine these two architectures to provide visitors with semantic-based contextual information using digital broadcasting as communication technology.

As stated in the hypothesis of this dissertation in Section 1.4, the objective is to filter all the incoming information by means of the rules defined on top of the ontology in order to only display on the screen of mobile devices the piece of information that is really relevant to tourists given their current context.

Considering only the digital broadcasting architecture shown in figure 5.2, the information shown in the mobile device on the client side without any further processing other than the one specified in the architecture, would correspond to all the information broadcasted, i.e. relevant and non-relevant information. This would not be of significant support for visitors, since they would have to actively browse all the information, looking for the particular piece they are looking for, not even exactly knowing what is that they are looking for. This would create a cognitive overflow and would discourage users to use a CONCERT-based application.

So, if an approach could be found to combine the digital broadcasting architecture in figure 5.2 with the CONCERT architecture in figure 5.1, there would be a way to further process tourism broadcasted information by means of context values. Thus, figure 5.6 presents the conceptual approach to the

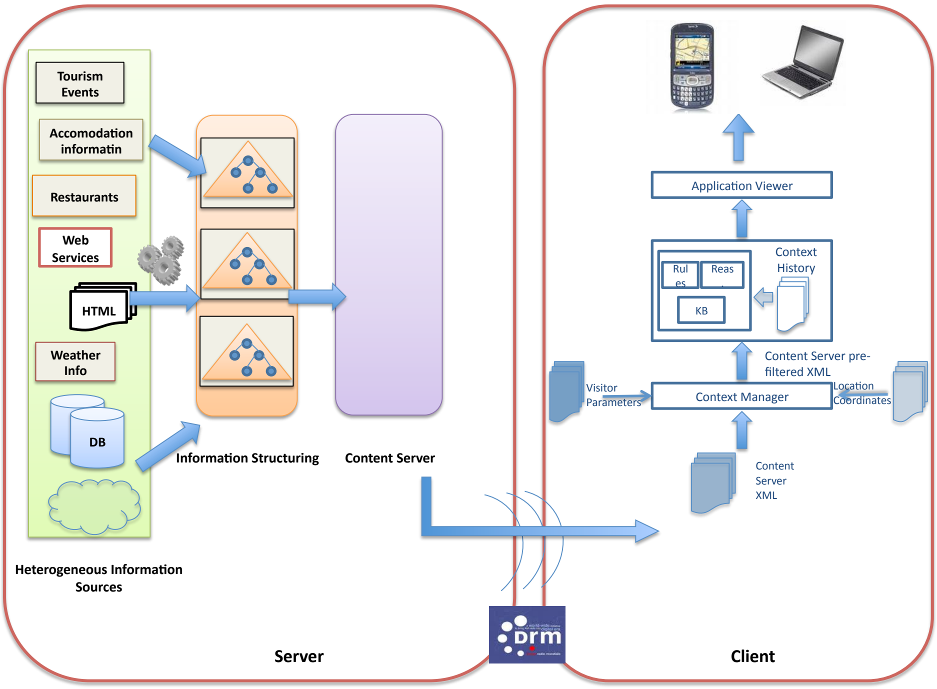

Figure 5.6: CONCERT digital broadcasting architecture

semantic-based broadcasted information filtering process that takes places with the combination of both architectures.

The CONCERT architecture seats on top of the broadcast architecture on the client side in another layer. It receives DRM signal and acts as a filter over the decompressed XML file received by the Journaline module through semantic processing automatically providing visitors with context dependent information. The details of how this architecture works are thoroughly described in the following Sections.

5.4 CONCERT based application for tourism

The conceptual approach presented in the previous Section is now thoroughly presented in practical terms. In order to answer the research questions and to validate the research hypothesis put forward in Section 1.4, a CONCERT framework-based application has been developed for laboratory experimentation. The following picture (figure 5.7) shows its architecture.

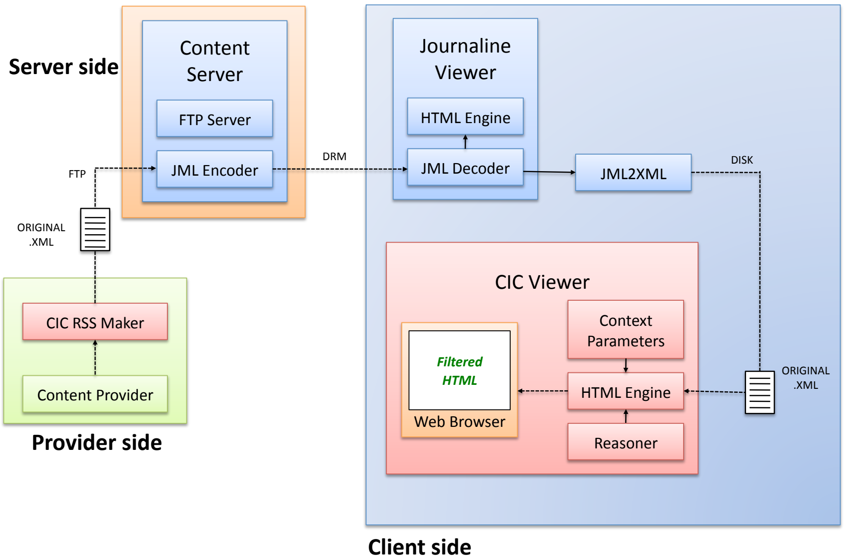

Figure 5.7: CONCERT framework based application architecture

The content provider uses a tool (CIC RSS Maker, Section 5.4.1) in order to create the content. By means of this tool, the content is uploaded to the Content Server (server side), further processed and compressed into JML and broadcasted. Once in the client side, the Journaline viewer decodes the incoming JML file and converts it back into an XML file in the JML2XML engine. The (converted) XML file (that is similar to the one uploaded to the Content Server by the content maker) is then processed by the network of ontologies and produces a final XML file, that is further processed into an HTML file, which is finally displayed in the mobile device's screen.

The real contribution and what the evaluation in Chapter 6 will look into, is to what happens with the information after it has been decompressed. Thus, figure 5.8 presents the sequence diagram and shows the sequence of events that take place at the CONCERT architecture, once the incoming information has been converted back into an XML file (Section 5.4.2).

The following subsections describe more in detail each of the components of the CONCERT-based application that has been developed in the framework of this dissertation.

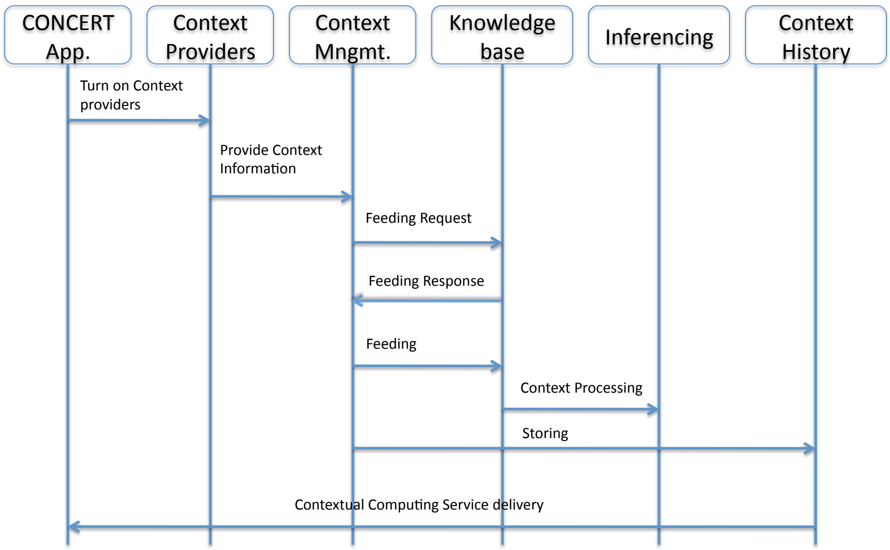

Figure 5.8: Sequence Diagram

5.4.1 CIC RSS Content Maker

As stated earlier, context and tourism information are either automatically gathered from the Internet and/or uploaded to the Content Server. Producing all the Web crawlers or search engines to automatically search, access and retrieve context and tourism related information from distributed and heterogeneous sources of information is beyond of the scope of this dissertation. However, a Content Maker tool, the CIC RSS Content Maker, for experimental purposes of the CONCERT framework-based application has been implemented.

As mentioned earlier, the CONCERT framework gathers information from the Internet. Information is actually gathered from external entities, which can be humans, services, or anything that provides the system with information that can later be broadcasted. In addition, DMOs or official tourism offices, as stated earlier, can also actively provide information. For example, typical information providers could be tourism information offices, destination management organizations, Websites with meteorological information, etc. For the sake of this dissertation, a tool called CIC RSS Maker has been developed with the purpose of creating tourism information to be later "broadcasted" in the laboratory (Fig.5.9).

This tool allows to create any kind of tourism information. The tool allows to introduce text-based information. This information is given the specified XML format and then uploaded via FTP to the Content Server installed in the mugiLab[8] laboratory in CICtourGUNE. The Content Server then emulates the DRM signal and "broadcasts" the tourism information.

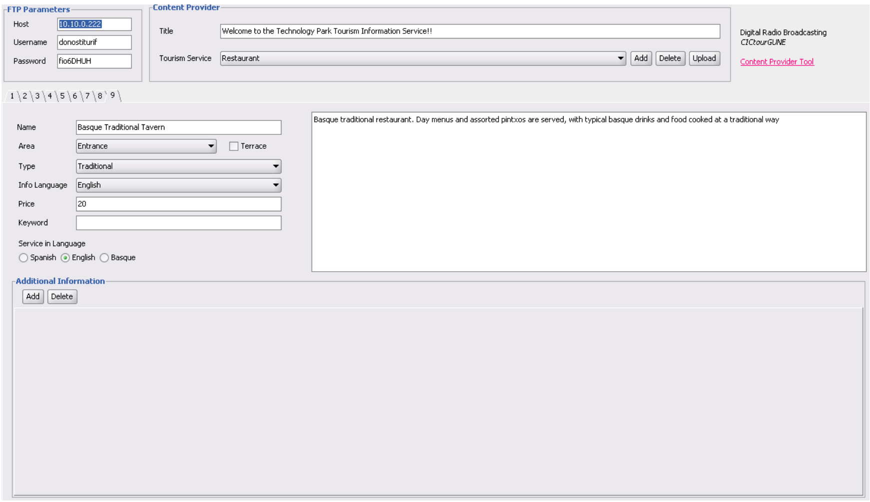

Figure 5.9: The RSS Maker tool

Figure 5.9 shows a screenshot of the CIC RSS Maker. The FTP connection parameters can be found on the upper left panel. These data indicate the IP address where the information generated by means of the CIC RSS Maker will be stored. In addition, user name and password are also provided so that direct interaction with the Content Server interfaces is avoided.

The Content Provider panel allows to give a title to the services that are going to be generated, in this case *Welcome to the Technology Park tourism*

[8] http://www.tourgune.org/en/5/0/0/0/mugilab.html

information service!! In addition, it allows to create a number of different kinds of tourism services, such as restaurants, hotels, museums, events, and so on. The "Add" and "Delete" buttons allow to create and remove the tourism objects as desired.

When pressing the "Add" button in the Content Provider panel, a new tab is created in the centre of the screen. The user only needs to complete the information as corresponds in terms of name of the tourism resource, location, type, information language and price. In case a new keyword (for the XML file) is required to be created, there is a text field for that purpose. On the text field on the right hand side, there is plenty of room to introduce the description of the tourism object that is being broadcasted.

At the bottom part of the screen, the "Additional Information" area is to be found. By pushing the "Add" button located in that part of the screen a new object is created, which may be accessed from the previous one, following the schema shown in figure 5.4.

This structure is determined by the ETSI JML Journaline, as specified earlier in Section 5.2.5. The kind of information that has been developed for the purposes of CONCERT evaluation is relatively simple, however, the Journaline Schema allows to create distinct structures of information, following the hierarchy shown in figure 5.5.

Each JML object can be associated with its corresponding language or geographic area, for example. These parameters can be customized according to the specific needs of the application to particular languages and locations, which are especially relevant variables for tourism. A geographic area may be given a logic name, e.g. San Sebastian Technology Park, which is translated in a physical value determined by a geographic location: a coordinate and a square defined by a distance "n" from the central coordinate, as shown in figure 5.10.

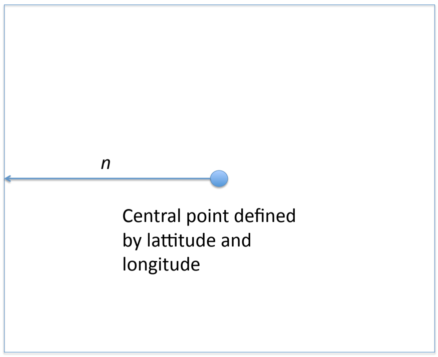

Figure 5.10: Area definition in the CONCERT framework

Once all the desired information has been provided to the system, by pressing the “Upload RSS” button, the information is uploaded to the content server. The CIC RSS Maker produces an XML document that fulfils the JML standard mentioned earlier.

```
<?xml version="1.0" encoding="UTF-8"?>
<journaline xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/
XMLSchema-instance">
<menu>
    <title>Welcome to the Tourism Information DRM Service</title>
        <message>
            <objectparameters>
                <defaultlanguage iso="eng"></defaultlanguage>
            </objectparameters>
            <title>
                Weather information
                <keyword description="Tourgune"></keyword>
            </title>
            <body>
                <text>Cloudy with occasional light rain...</text>
            </body>
        </message>
</menu>
```

Figure 5.11: An example of an XML file

An example of an XML file created by the CONCERT Content Maker is shown in figure 5.11. As can be noted in the figure, there are various keywords in order to format the content. These keywords are defined by the ETSI JML Journaline guidelines. Interestingly, the JML standard provides by default with a label called *defaultlanguage*, which allows to match an object with a language. At the same time, in order to relate a particular geographic location, the label *keywords* is going to be used. This label associates key words with JML objects through geographic logic names.

Picture 5.12 shows the standard labels provided by the JML standard to denote geographic locations. However, for the sake of processing simplicity, the matching will be carried out by means of the logic names as mentioned, together with the label *keywords*.

```
<objectparameters>

    <georegions>

        <region>

            <position lon="-73.973256" lat="40.764393"/>
            <position lon="-73949319" lat="40.796868"/>
            <position lon="-73.958176" lat="40.800632"/>
            <position lon="-73.982112" lat="40.767850"/>

        </region>

    </georegions>

</objectparameters>
```

Figure 5.12: An example of coordinates keywords

5.4.2 Journaline viewer and JML2XML engine

In order the ContOlogy network of ontologies to process incoming information by means of the rules, the information has to be processed in advance when received by the client in the Journaline. When the information is received in the client, the following two cases can occur:

- *Journaline Viewer*: As indicated at the end of Section 5.2.2, the Journaline Viewer decodes the incoming JML file broadcasted by the

Content Server in the client side. JML files are decompressed into HTML files by the JML Decoder and it saves the files in a known path. These files can be then displayed in a traditional Web browser, such as the Mozilla Firefox or Internet Explorer on a Windows Operating System Machine. It is important to remark at this point, that information viewed this way has not been processed by the CONCERT framework and therefore, the HTML files have not been filtered by the context parameters defined by CONCERT.

- *JML2XML*: The information can be further processed when received by the Journaline module. In order to perform that transformation, a specific module has been developed by the Fraunhofer IIS *ex profeso* for the purposes of the experimentation carried out within the CONCERT framework. It converts the incoming JML files into a unique XML file which, at the same time, is equivalent to the one created by the CIC RSS Maker. Therefore, the JML2XML module allows CONCERT to make use of all the rough information as uploaded to the Content Server, in the server side, and enables the context manager to process all incoming tourism and context information as described earlier.

5.4.3 Semantic-based filtering engine: rule and ontology based reasoning

Throughout the entire dissertation, ontologies have been claimed to be a crucial tool not only in general for contextual computing frameworks and/or applications, but also, for the particular case of the CONCERT framework. Chapter 3 has first demonstrated the need for a new semantic approach for context management in the form of networks of ontologies and second, it has justified the use of ontologies for a number of reasons. Then, Chapter 4 has described in detail the development process of the ContOlogy network of ontologies. It is now time to describe how the network of ontologies works, i.e. how it checks the consistency of the context model and then selects (filters) incoming tourism information by means of rules with base on current context values.

When taking a formal approach to model context, context can be processed with logical reasoning mechanisms. The use of context reasoning has two folds: Firstly, checking the consistency of context and secondly, deducing high-level implicit context from low-level explicit context[WZGP04].

Consistency check is crucial in the definition of an ontology as well as in its population. Indeed, when the ontology is populated with instances obtained from various context sources through the context manager, consistency checking is performed in order to capture possible inconsistencies (e.g., the same instance belonging to disjoint classes, a person localized in different locations at the same time)[ABR05].

Context reasoning is basically performed to derive higher levels of context from lower levels of context[GWPZ04][Nic08]. For example, if a temperature sensor measures 0 Celsius (low level context, Section 3.1.3), then the system may infer that it is cold (high level context). There are a number of different reasoning approaches in context-awareness, e.g. by application, i.e. hard coded, probabilistic, i.e. Bayesian Networks[GPZ04], or rule-based with OWL ontologies. Context may be defined as facts, expressions and rules and has a high degree of formality, therefore, its model is often based on ontologies.

The existing proposals for building a rule layer on top of the ontology layer of the Semantic Web refer to rule formalisms originating from the field of Logic Programming[ADG+05]. Logic defines conditions on which a concluding expression or fact may be derived from a set of other expressions or facts, i.e. reasoning.

In particular cases, contextual data can be derived through ontological reasoning only populating the ontology with information provided by different entities. In those cases, reasoning must be performed on-demand at the time of the service request. Reasoning is performed in real-time and is based on both description logic and user-defined logic rules. However, this approach is unsuitable for time- critical applications[WZGP04][ABR05], for rule processing usually lasts several seconds (see Chapter 6).

A more flexible reasoning mechanism is user-defined reasoning. Through the creation of user-defined reasoning rules within the entailment of first-order logic, a wide range of higher-level, conceptual context such as what the user is doing can be deduced from relevant low-level context. The equivalence of OWL and description logic allows OWL to exploit the considerable existing body of DL reasoning to fulfil important logical requirements. These requirements include concept satisfiability, class subsumption, class consistency, and instance checking[WZGP04].

Considering the previous, the ContOlogy network of ontologies has been developed not only to provide the CONCERT framework with interoperability at the model level, but also, to be able to check the model's consistency and infer higher level contexts. In addition to that, the rules that have been defined on top of the network of ontologies allow to filter incoming broadcasted information according to certain context values. The network of ontologies has been developed with the open source Protégé ontology editor, version beta 3.4.1. This editor incorporates internal reasoners that enable the ontology consistency checking at the time of development. During the development of the ContOlogy network of ontologies, the Pellet 1.5.2 internal reasoner was used to check the ontology's consistency. In addition, in order to evaluate the ontology inference capabilities, several SWRL[9] rules have been defined. The JESS reasoning engine[10] has been used to run the rules and prove that the ontology was actually reasoning as expected by design.

However, the network of ontologies cannot work independently and it needs to be integrated within the rest of the architecture presented in figure 5.1 in order to provide the CONCERT framework full functionalities. The rest of the components of the CONCERT based application depicted in figure 5.1, have been developed in Java[11] for evaluation and assessment purposes of the framework of this dissertation. The Eclipse open source developing environment version 1.7.0.v2 has been used. Figure 5.13 shows the relation of projects, packages and classes developed in order to implement the CONCERT architecture.

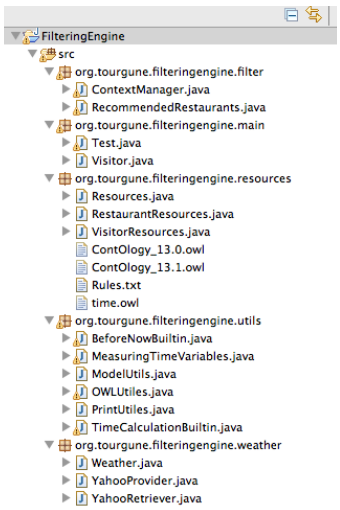

Figure 5.13: The Java directory of projects, packages and classes

The functionality implemented by each of the components is shown in table 5.3.

[9] http://www.w3.org/Submission/SWRL/
[10] www.jessrules.com/
[11] http://java.sun.com/

Package	Functionality implemented
Filtering Engine	All context information management, Jena models of the ontology, reasoning. Loading tourism objects
Main	The main.java class, where the execution begins and the objects and classes are instantiated
Resources	Where the basic resources reside, i.e. ontology, rules, java classes managing ontology content
Filtering Engine	All auxiliary classes such as extra built-ins, validation (consistency) of the model, printing, displaying, time measuring
Weather	The Web service that connects with the, in this case, Yahoo! Weather service

Table 5.3: Functionality of Filtering Engine java classes

In order to manage semantic Web-based applications in Java environments, there are various Java APIs, such as Pellet, Racer or Jena. One widely adopted option is Jena. Jena is a toolkit for developing applications within the Semantic Web. Figure 5.14 shows schematically how it works. The Jena inference engine is designed to allow a range of inference engines or reasoners to be plugged into Jena. Such engines are used to derive additional RDF assertions which are entailed from some base RDF together with any optional ontology information and the axioms and rules associated with the reasoner. JENA has been adopted for the following reasons: It has a Java API that allows to manage ontologies. In addition, it allows to use a larger number of *built-ins* and in addition to that, tailored *built-ins* can be programmed according to the needs of specific rules.

The primary use of this mechanism is to support the use of languages such as RDFS and OWL, which allow additional facts to be inferred from instance data and class descriptions. However, the machinery is designed to be quite general and, in particular, it includes a generic rule engine that can be used for many RDF processing or transformation tasks.

Applications normally access the inference machinery in JENA by using the ModelFactory (this is the case in the ContextManager.java of CONCERT) to associate a data set with some reasoner to create a new model. Queries to the created model will return not only those statements

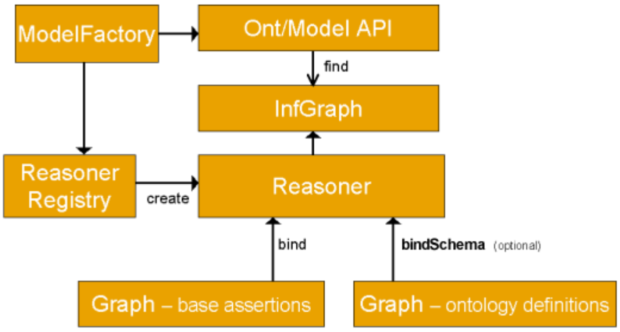

Figure 5.14: The Jena Structure

that were present in the original data but also additional statements than can be derived from the data using the rules or other inference mechanisms implemented by the reasoner. For the purpose of the CONCERT framework, the OWL DL reasoner was used.

In particular, the Ontology API provides convenient ways to link appropriate reasoners into the OntModels that it constructs. As part of the general RDF API we also provide an InfModel, this is an extension to the normal Model interface that provides additional control and access to an underlying inference graph.

The reasoner API supports the notion of specializing a reasoner by binding it to a set of schema or ontology data using the bindSchema call. The specialized reasoner can then be attached to different sets of instance data using bind calls. In situations where the same schema information is to be used multiple times with different sets of instance data then this technique allows for some reuse of inferences across the different uses of the schema. In RDF there is no strong separation between schema (aka Ontology aka tbox) data and instance (aka abox) data and so any data, whether class or instance related, can be included in either the bind or bindSchema calls - the names are suggestive rather than restrictive.

To keep the design as open ended as possible Jena also includes a ReasonerRegistry. This is a static class through which the set of reasoners currently available can be examined. It is possible to register new reasoner types and to dynamically search for reasoners of a given type. The ReasonerRegistry also provides convenient access to prebuilt instances of the main supplied reasoners.

The available reasoners in Jena are the following:

- *Transitive reasoner*: Provides support for storing and traversing class and property lattices. This implements just the transitive and reflexive properties of rdfs:subPropertyOf and rdfs:subClassOf.

- *RDFS reasoner*: Implements a configurable subset of the RDFS entailments.

- *OWL, OWL Mini, OWL Micro Reasoner*: A set of useful but incomplete implementation of the OWL/Lite subset of the OWL/Full language.

- *DAML micro reasoner*: Used internally to enable the legacy DAML API to provide minimal (RDFS scale) inference capabilities.

- *Generic rule reasoner*: A rule based reasoner that supports user defined rules. Forward chaining, tabled backward chaining and hybrid execution strategies are supported.

Jena includes a general purpose rule-based reasoner which is used to implement both the RDFS and OWL reasoners but is also available for general use. This reasoner supports rule-based inference over RDF graphs and provides forward chaining, backward chaining and a hybrid execution model. To be more exact, there are two internal rule engines one forward chaining RETE engine and one tabled datalog engine. They can be run separately or the forward engine can be used to prime the backward engine which in turn will be used to answer queries.

The various engine configurations are all accessible through a single parameterised reasoner GenericRuleReasoner. At a minimum a GenericRuleReasoner requires a ruleset to define its behaviour. A GenericRuleReasoner instance with a ruleset can be used like any of the other reasoners. i.e. it can be bound to a data model and used to answer queries to the resulting inference model.

The rule reasoner can also be extended by registering new procedural primitives. The current release includes a starting set of primitives which are sufficient for the RDFS and OWL implementations but is easily extensible.

Considering all of the previous, the following use case scenario illustrates the reasoning functionality of the CONCERT context framework. Let us consider a visitor (Carlos) that is participating in a conference. Carlos turns his CONCERT based application on and introduces his personal information in the framework through the CONCERT viewer application, Section 5.4.4. That information will later be used as one of the three context providers when the application runs. Once the morning sessions are finished, Carlos is having a lunch meeting with an important client, so he needs to find a good restaurant in the area. However, after having checked the context history, the CONCERT application determines that it is the first time Carlos is in that city and does not know which restaurant would be adequate. The CONCERT-based application knows where Carlos is located through the location coordinates provided by the embedded GPS sensor and also knows not only about the business lunch, but also about the amount of time available for lunch by reading Carlos' agenda in the mobile device.

The context manager starts to run and gathers all the contextual information, i.e. Carlos' personal preferences, location coordinates from the location sensors embedded in the mobile device and the Journaline XML file, converted by the JML2XML. The context manager then filters the XML by location and produces another XML that contains only information about Carlos' location. Following, the context manager checks the status of the knowledge base: if it has some content from previous use, it removes it from the knowledge base and stores it in the context history repository and then, it fills the knowledge base with the new XML file together with Carlos' personal information into the knowledge base. Otherwise, the mapping is done straightforwardly. The mapping process is completed through matching predefined keywords of the incoming XML file as individuals in their corresponding classes within the ontology in order to simplify data traffic between the context manager and the knowledge base.

Once the situational context (i.e. the instances) has been loaded on the knowledge base, the rule-based reasoning process begins. The inference engine runs the rules that have been defined (example of such rules can be found on table 5.4) on the network of ontologies and produces the results.

These rules have been defined following the Jena syntax and in forward mode. This means that if and only if all the antecedents are true, then the consequent is also true and the associated action will be triggered.

5.4.4 CIC Viewer

The CIC viewer has been implemented in order to be able to visualize broadcasted incoming information. It allows to implement the CONCERT framework's various functionalities. It is hosted in the client side, i.e. in the mobile device. The CIC viewer has been specifically built to carry out the user evaluation, presented in Section 6.4.

Rules
(?v rdf:type dcl:Visitor), (?v dcl:hasMotivation dcl:AttendingMeetingMotivation), (?v dcl:hasPreferences dcl:English), (?r rdf:type dcl:ModernRestaurant), (?r dcl:hasTerrace "false"^^xs:boolean), (?r dcl:offersServiceInLanguage dcl:English), (?r dcl:hasPrice ?p), lessThan(?p,12), (?m rdf:type dcl:Activity), (?v dcl:isDoing ?m), (?e rdf:type time:TemporalEntity), (?m dcl:hasTemporalEntity ?e), (?f rdf:type time:Instant), (?e time:hasEnd ?f), (?f time:inXSDDateTime ?z), (?v dcl:hasScheduled ?n), (?n dcl:hasTemporalEntity ?u), (?u time:hasBeginning ?b), (?b time:inXSDDateTime ?y), beforeNow(?y) -> (?r dcl:isTourismServiceOfferedToVisitor ?v)

Table 5.4: An example of the Jena rules used in CONCERT

The CONCERT viewer in figure 5.15 represents the interface a potential CONCERT-based commercial application would have. It is divided into two parts. On the left hand side of the screen, the Location, Visitor and Today's Agenda panels allow to introduce visitor parameters. The GPS sensor embedded in mobile devices provides automatically visitors' location. In case there was a failure, location information can be manually typed into the application in the "Location" panel.

The "Visitor" panel allows to provide personal information. This panel has been developed following the keywords of the JML file and the structure of the ContOlogy network of ontologies, which at the same time, has been built based upon the guidelines provided by the UNWTO for definition of visitors[UNW08a]. This way, the user can introduce its preferred language, motivation and for the sake of the prototype example, food preferences. Finally, the Today's Agenda panel allows to introduce the scheduled activities for the user. These activities can also be automatically read from the mobile

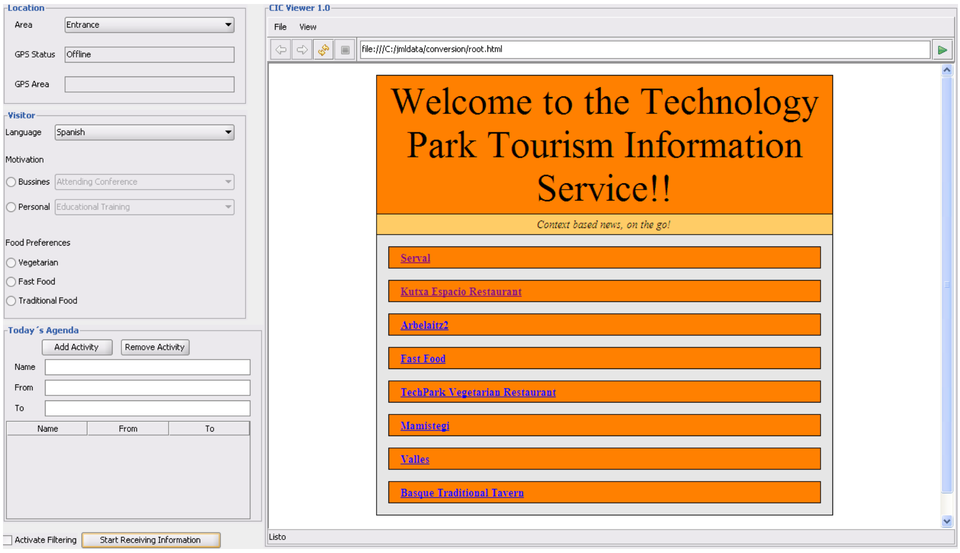

Figure 5.15: CONCERTviewer

device agenda. However, in case no activities have been read, the user can provide them manually. One of the objectives of this is to try to detect time constraints. For example, if somebody is a *Attending Meeting Visitor* and has only one hour for lunch, the CONCERT-based application would display preferably fast food restaurants. There are multiple options and potential recommendations depending on the motivation of the journey, role played by visitors and time constraints.

On the right hand side of the screen, the viewer shows a Web browser-based kind of display. That part of the viewer is responsible for showing the incoming information. By clicking on the name of the tourism objects, one may access a more detailed information about them, following the structure shown in Section 5.2.5 for JML objects. The right and left arrows on the top of the screen allow to navigate backwards and forward, and the other button, allows to refresh the screen.

When pushing the “Start Receiving Information” button, all the broadcasted information received by the mobile device will be shown to the visitor in the Web browser-like display. On the bottom left hand side, there is a tick box, “Activate Filtering”. By clicking the tick box, the reasoning engine begins to run and the whole functionality of the ontology is implemented. The rules defined for the different scenarios are performed and the resulting subset of relevant tourism information is also displayed on the right hand side of the screen.

5.4.5 CIC Viewer’s workflow

The following pictures show the sequence of the screenshots visitors would see on their devices when activating the CONCERT framework-based application.

1. Turning on the system (fig. 5.16).

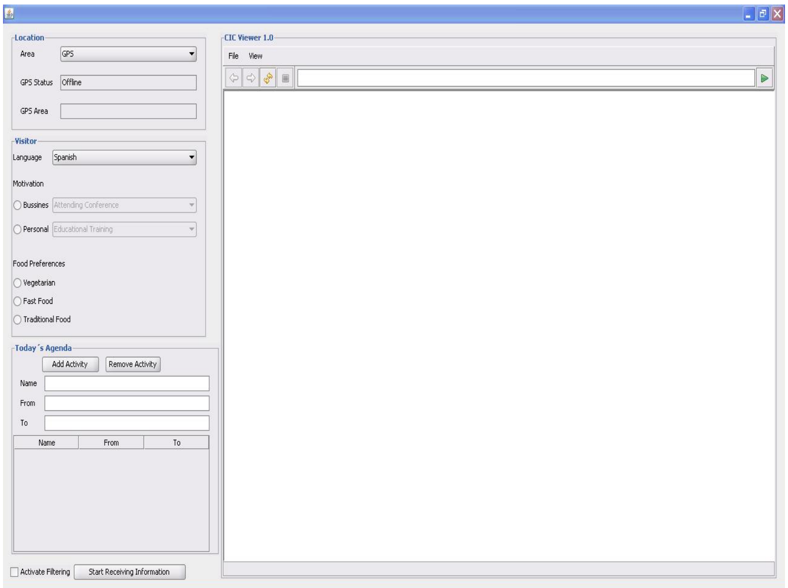

Figure 5.16: Turning on the System.

2. Introducing personal information into the system (fig 5.17).

3. Pushing the “start receiving” (information) button (fig 5.18).

4. Activating the semantic filter and obtaining just context-based (relevant) information (fig. 5.19).

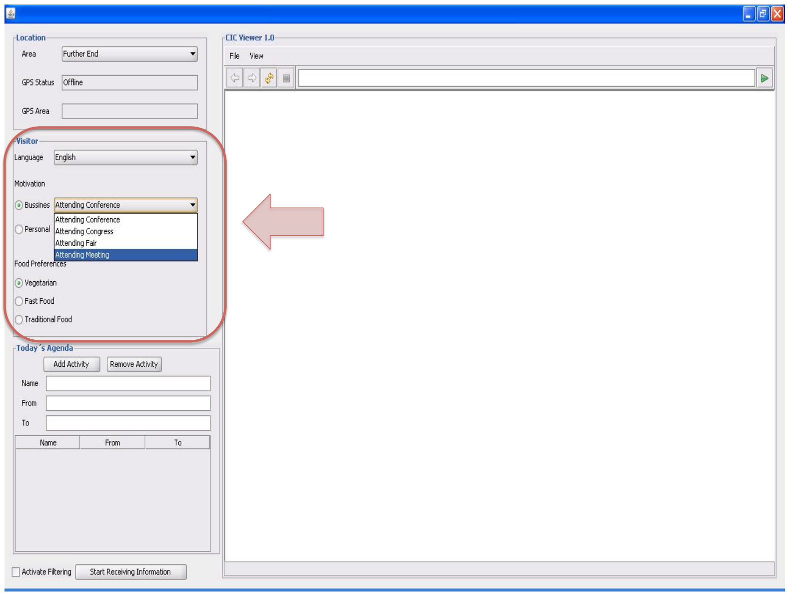

Figure 5.17: Introducing personal information.

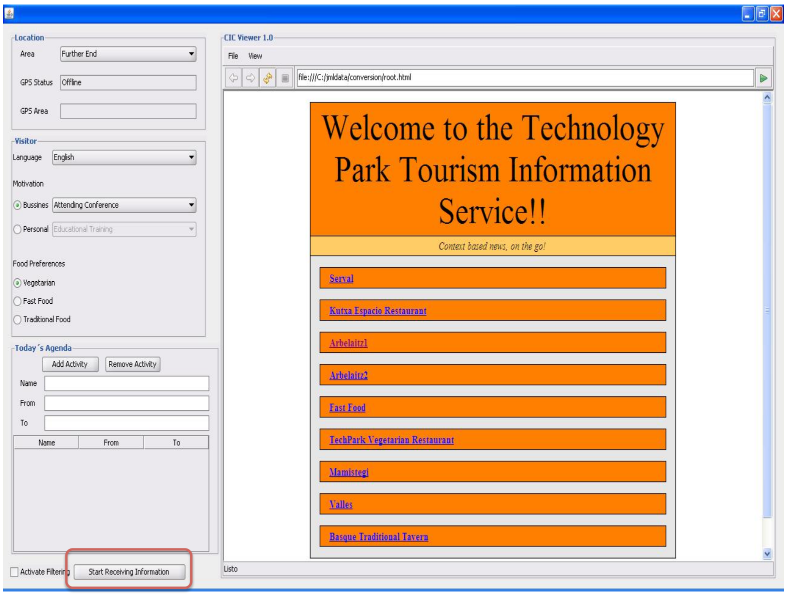

Figure 5.18: The System receiving information.

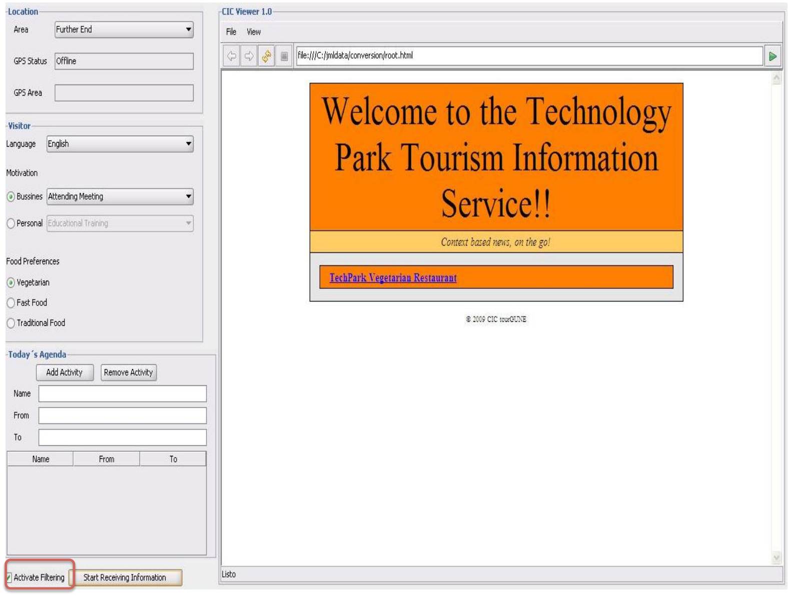

Figure 5.19: Displaying only context based information.

5.5 Summary

The need for architectural support is essential in any computer science application. Even so more in context-aware applications, as demonstrated throughout this entire Chapter, given the number of distinct and distributed components that integrate them. Chapter 5 represents the conclusion, the realisation and implementation of what has been claimed in Chapters 3 and 4. Chapter 3 has provided the theoretical foundations of this dissertation, Chapter 4 has provided the computational model and, finally, Chapter 5 has provided the implementation base for the theory to become reality.

Throughout the whole chapter, the different sections have guided towards the semantic-based contextual computing digital broadcasting tourism service developed in this dissertation. The implications of the theoretic contributions explained in Chapter 3 have been the base upon which the CONCERT architecture has been built (Section 5.1). It is a rather simple contextual computing architecture, yet considering the minimum amount of elements and interconnections among them. However, it is remarkable at the same time its independence from the basic context gathering and dissemination infrastructure, which makes it potentially usable in other context-aware infrastructures.

One of the particularities introduced by the CONCERT framework is the use of digital broadcasting for context and tourism information dissemination from context sources to the client. This has a number of advantages, e.g. coverage, cost (especially under roaming conditions), clear and solid information structure that allows a simple navigation, etc. On the contrary, it also presents a number of disadvantages that need to be specified at this point, e.g. no up-link digital broadcasting communication channel available to make queries and no receivers adapted for mobile phones (there are some for laptops), etc. However, given the popularity of this technology in the other sectors, such as the automotive, and its potentiality in the realm of tourism, this dissertation experiments with DRM aiming at opening the path for future tourism broadcasting services, both in general and for specific *subsectors*, such as trekking and nautical maritime sectors. Thus, Section 5.2 has presented the different radio broadcasting standards and has justified the use of the DRM standard for this dissertation.

The Chapter has continued in Section 5.3 by showing a conceptual approach to a semantic-based contextual computing architecture in tourism. The architecture shown in figure 5.6 is a combination of both previous architectures, i.e. the CONCERT architecture in figure 5.1 and the broadcasting architecture 5.2. The CONCERT architecture lays on top of the digital broadcasting architecture and filters incoming information by means of context values.

Finally, Section 5.4 shows a practical implementation, thoroughly describing and analyzing not only the components of the CONCERT-based application developed in this dissertation, but also analyzing their behaviour and functioning in order to meet the objectives shown in Section 1.5.

Having stated all of the above, this Chapter is even so more relevant because it answers the two main research questions. The research work carried out throughout this dissertation is based upon two research questions that are put forward in Section 1.4. Following the research questions, a hypothesis (that has been divided into different sub-hypotheses for clarity) is suggested. These then lead to the three general objectives.

The research questions and the first and second sub-hypotheses are enumerated following. Their coverage in terms of the implementation of the application, based on the theory developed in Chapter 3, is shown:

- The first research question inquires about *How can contextual computing information based applications be implemented with minimal infrastructure and deployment overload?* Indeed, the architecture shown in Section 5.1 has almost no infrastructure dependency. On the one hand, it does not depend on any physical sentient environment infrastructure, for the way contextual information is gathered. On the other hand, the infrastructure needed to implement the CONCERT framework architecture is minimal and is the standard one provided by any programming environment, such as Eclipse.

The architecture shown in Section 5.1 minimizes the developing efforts required for minimum contextual computing applications.

- The second research question looks into the following: *Can ontologies be used to filter incoming push broadcasted information according to the values of context at a given location and time?* Section 5.4 describes a CONCERT based application. This application uses ontologies in the following way: the values of context are instantiated in the ontology and used as reference values to filter incoming broadcasted information. Rules defined on top of the ontology take these instances and compare them to the ones that characterize the incoming tourism objects, effectively filtering all the information and providing the user only with relevant tourism information. The second research question is the corroborated as follows.

Therefore, ontologies can be used to filter push broadcasted information according to the values of context at a given moment of time.

In addition to this, the application developed in the framework of this dissertation and presented in Section 5.4 corroborates the first two sub-hypothesis presented

- Sub-hypothesis 1 claims *A push contextual computing framework can be established by gathering context information from Internet information sources without external sensors.*

Indeed, the CONCERT framework receives context information from the embedded GPS sensor and from Internet sources of information, such as weather. It also receives information regarding the user from the user interface, i.e. the CONCERT Viewer.

- Sub-hypothesis 2 claims that *Digital broadcasting can be used to disseminate context and tourism information to mobile devices equipped with a filtering engine designed to support human mobility.*

The experiments that have been done with the CONCERT-based application described in Section 5.4 demonstrate that digital broadcasting can be first disseminated and then filtered with a semantic engine.

Chapter 6

Evaluation of the CONCERT Framework

"Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience".
Roger Bacon, (1214 - 1294), Doctor Mirabilis, English philosopher and Franciscan.

Experimentation is central to the scientific process[Tic98]. Traditional sciences, such as physics, biology or astronomy, for example, have widely used experiments for theory testing and exploration. On the contrary, there is a big debate going on whether computer science is really a science comparable to traditional sciences and if it is the case, on how much is experimented within the realm [Den05][Den03][Den81][DCG+89][Fei06][HMP04][Ros98][Tic98]. This debate is obviously far beyond the scope of this dissertation. However, there are a number of facts that are worth mentioning.

There are two principles that underlie computer science, namely mechanics and design[Den03][HMP04]. Mechanics deal with the structure and operation of computation elements, whereas design enables computer professionals to harness (computer) mechanics in the service of users and customers. In fact, design is one of the three paradigms (the other two are theory and abstraction), in which computer science rests[DCG+89]. Design is rooted in engineering and suggests to follow four steps when it comes to the construction of a system (the CONCERT framework-based application, for example) to solve a given problem, i.e. provide answers to the general research questions and corroborate (or refute) the hypotheses that have been put forward:

- *State the requirements*: The new theoretic approach to the notion of context suggested in Chapter 3, the new definition of context derived from it, its scope and the particularities of the CONCERT framework have set the requirements for such an application to be built.

- *State specifications*: The specifications of the system are derived from the focus that drives the notion of context in the framework of human mobility (Chapter 3), and the implications of that definition and scope of the context model built in Chapter 4.

- *Design and implement the system*: Once the theory and model have been defined and designed, the system and/or applications need to be implemented to check whether the theory is corroborated by practice or, on the contrary, it is refuted. As stated in the previous Chapter, by just implementing the CONCERT framework-based application the two research questions as well as the first two sub-hypotheses (H1 and H2) have been corroborated.

- *Test the system*: However, there are still another two sub-hypotheses that need to be tested. Therefore, the system has to be assessed in technical terms to analyse and experience how it actually works, what its performance is and to find out critical variables amongst other issues. The information derived from evidence extracted from these observations could shed light to visualize other kinds of systems, improvements on the existing application or other research areas.

  The test of the system extends also technical aspects. In fact, the fourth sub-hypothesis presented in Section 1.4 revolves around the fact that the use of CONCERT increases user satisfaction because of the characteristics of the context ontology and how it filters incoming information. Due to this reason, real people have to be involved in the evaluation of the CONCERT framework-based application.

Hence, after having elaborated the theory that underlies the CONCERT framework-based application and implemented the system itself, it is now time to experiment. Having said this however, the ContOlogy network of ontologies has been validated separately in Section 4.7. In other words, before integrating the model within the whole framework, the correctness of the network of ontologies has been guaranteed. The evaluation, therefore, has been carried out gradually at different stages.

Then, the objective of this chapter is not only to understand how the theory developed in the framework of this dissertation works in practice, but also, to test the system in terms of performance analysis and user satisfaction.

In addition, the evaluation work and its subsequent analysis should help elucidate new areas of research[Tic98], that will be specified in Chapter 7.

Chapter 1 presented the two research questions and the hypothesis on which the rest of the dissertation has been carried out. The research questions and the first two sub-hypotheses have been proved true in the previous Chapter, as stated earlier. The other two sub-hypotheses need other kinds of activities in order to verify them:

- *Technical Evaluation*: To test the performance of the piece of software that has been developed in terms of memory consumption, processing time and the relation of the variables, amongst others. The results of this experiment ought to corroborate the third sub-hypothesis of this dissertation, i.e. *A network of ontologies approach can be used to model the context of a visitor and use it as tourism information filtering engine based upon DL and rule based reasoning.*

- *User Evaluation*: To measure the perceived usefulness of the application by real people, i.e. visitors. The results of this experiment ought to corroborate the fourth sub-hypothesis of this dissertation, i.e. *This framework increases user satisfaction by using a semantic-based visitor-centred model for context, which is built following the recommendations on visitor classification established by the UNWTO.*

In this manner, the Chapter begins in Section 6.1 by describing the evaluation scenarios in which the application is going to be examined. They are abstractions of situations that may take place in real life. Due to the technology used, i.e. digital broadcasting, these situations have "virtually" been recreated in the laboratory environment. The application has been run under these assumptions and its performance measured.

The following Section, Section 6.2, will deal with the characteristics of the prototype to be employed by the participants in the user evaluation. This Section is based in a great extent upon Section 5.4, and explains with more detail the kind of information the application processes, how it receives broadcasted information and the reasoning process.

Section 6.3 will deal with the Technical Evaluation. This section is devoted to the realization of experiments in different machines in order to analyse the functioning of the CONCERT-based application. The variables that are monitorized are described and then, considering the particularities of the application (derived from its own nature in terms of theoretic base, technologies used and limitations) the experiments will show how the system behaves on the different scenarios defined in Section 6.1, its performance and the relation (if there is any at all) amongst the variables. As mentioned earlier, the results provided by these experiments will be used as evidence to decide whether the third sub-hypothesis has to be corroborated or refuted.

Besides the scientific, research and technological challenges this dissertation faces, the final objective of CONCERT is not only to provide a theoretical framework, but also to boost a widespread adoption of contextual computing-based applications in general, and in tourism in particular. Investigating the factors associated with user acceptance of new software systems has been an important research stream in the field of information systems[Che06]. There are several technology acceptance models, which measure in one way or another the perception and adoption of technology by people. Thus, Section 6.4 presents the whole process of the user evaluation carried out in the framework of this dissertation with real people.

Finally, the Chapter closes in Section 6.5 with an analysis of the results obtained in the two evaluation processes performed, as well as with a reflection on experimentation.

6.1 Evaluation Scenarios

The San Sebastian Technology Park has been equipped with a digital broadcasting service, whereby all tourism information concerning the park is continuously being broadcasted according to DRM (Digital Radio Mondiale) standards through the MDI (Multiplex Distribution Interface) protocol, as described in Chapter 5. This information contains data about restaurants and museums located at the Technology Park.

Prior to broadcast, the Technology Park's tourism authorities have classified the restaurants by location and type. They have then built an XML file containing that information following the requirements of the local broadcasting based Mobility Assistance system, whereby restaurants information is leveraged by means of pre-established keywords. They have uploaded the XML file to the Park's Tourism Service Content Server, converted it there to JML (Journaline Markup Language) format and broadcasted. They have used the CIC RSS Content Maker tool explained in Section 5.4.1 for this purpose.

The park has been (virtually) divided into three parts namely Entrance, Main Building Area and Further End. The restaurants to be found in the park, classified by location and type are the following:

- *Entrance*: two restaurants can be found at the entrance of the park.

  - *Basque Tavern*: This is a Basque traditional cuisine restaurant. This restaurant serves traditional *pintxos*[1] and sandwiches, together with day-menus based on home made vegetables, fish, meat and so on, all cooked in the traditional way. The whole menu costs around EUR 20 per person.
  - *Serval*: This is the second restaurant located at this end of the Technology Park. It is an ordinary day-menu restaurant. It serves dishes of the day or complete menus at a medium price. The menu per person costs EUR 15.

- *Main building area*: three restaurants can be found within the area of the main building:
  - *Kutxaespacio*: This is a traditional museum cafeteria/restaurant. It is by the Science museum and it offers some *pintxos*, sandwiches and day-menu dishes in prices between EUR 10 and EUR 15.
  - *Arbelaitz 1*: This is a elegant restaurant located in the main building of the park. This is a high quality and expensive restaurant where someone would take an important meeting for lunch. Day menus cost EUR 30 per person and regular menus cost around EUR 50 per person.
  - *Arbelaitz 2*: This is the cheaper version of the previous restaurant. Day-menus are served at a lower price and reasonable quality. It is located at the bottom of the main building. Menus are worth EUR 10.

- *Further part*: two more restaurants can be found at the furthest part of the Technology Park:
  - *Fast food restaurant*: A traditional fast-food restaurant of the McDonald's type. Sandwiches, hamburgers, chips, and hot-dogs are served in this restaurant and it is cheap. Menus at this restaurant cost around EUR 7 – 8.
  - *Vegetarian fast food restaurant*: A vegetarian restaurant serving only vegetables and salad and all natural food. It is also a fast food restaurant with medium price. Menus at this restaurant cost around EUR 7 – 8.

The following table shows a summary of the restaurants:

[1] http://en.wikipedia.org/wiki/Pincho

Location	**Restaurant Name**	**Type**	**Description**	**Price**
Entrance	Serval	Day-Menu	Regular Day-Menu Restaurant	EUR 15
Entrance	Basque Traditional Tavern	Traditional	Local Traditional Cuisine	EUR 20
Main building	Arbelaitz 1	elegant	High Quality Cuisine	EUR 30
Main building	Arbelaitz 2	Day-Menu	Regular Day-Menu	EUR 12
Main building	Kutxa Espacio	Cafeteria, Restaurant	Outdoor, museum restaurant	EUR 10 - 15
Further End	McDonald's	Fast Food	Fast Food	EUR 8
Further End	FresCo	Fast Food	Fast Food Vegetarian	EUR 8

Table 6.1: Summary of the restaurants located at the Technology Park

So, given the kind of services that can be offered at the Technology Park, the functionality of the CONCERT framework will be evaluated (first technically and then by real users) in the following scenarios.

1. *Scenario 1*: Businessman attending a conference.

   Carlos is a businessman attending a conference at the San Sebastian Technology Park about tourism and technologies. The conference takes place at the Park's main building and it begins at 9 am. Carlos arrives alone to the conference and takes a place in the auditorium.

   The conference finishes at about 1 pm and it is time for lunch. Carlos has a GPS and digital radio enabled mobile device that receives information from the Park's main broadcasting service. The context manager receives location information, broadcasted information and information about Carlos. Since Carlos is attending a conference, the system determines Carlos' role and gives it the value of *"AttendingConferenceVisitor"*, by reading it in Carlos' device calendar. The context manager further pre-processes the XML that has been broadcasted by location information and obtains information relative to Carlos' location. It finally sends all of this information to the Context Ontology.

Because it is lunch time, due to Carlos' location (main building of the Technology Park), the fact that Carlos has a meeting scheduled for 3 pm that day and that Carlos wishes to receive all of the information in English, the system provides, after reasoning, information about the Arbelaitz 2 restaurant.

2. *Scenario 2*: Businessman attending conference and working-lunch.

   Carlos is a businessman attending a conference at the San Sebastian Technology Park about tourism and technologies. The conference takes place at the Park's main building and begins at 9 am. Carlos arrives alone to the conference, but he is planning to have a business lunch (scheduled in Carlos' mobile device agenda) with a colleague he is meeting at the conference.

   When the conference finishes at 1 pm, Carlos meets his colleague. The system knows where Carlos is through his mobile device's GPS coordinates. The system also has information about Carlos' motivation (*"AttendingConferenceVisitor"*), preferences and motivation of the journey. In addition, Carlos wishes to receive all of the information in English.

   All of this information is sent to the application context manager, together with the XML file that has been broadcasted. The context manager pre-processes the XML according to location information and sends the result to the context ontology. Through reasoning, since Carlos is having a business lunch meeting after the conference the system finally recommends the Arbelaitz 1 restaurant, which would be appropriate in these cases.

3. *Scenario 3*: Worker looking for fast food.

   Carlos is a researcher attending several meetings at the San Sebastian Technology Park. It is 1 pm and he has just finished a meeting and has another one scheduled for 2 pm. He is not familiar with the area and he has no much time for lunch.

   Carlos is using a mobile device that is connected to the Technology Park digital broadcasting service where he had his last meeting. Because he is attending a meeting, the system determines that he is an *"AttendingMeetingVisitor"*. Carlos has specified in his preferences that he likes fast-food and he has no further preferences. All of this information, together with the XML file that has been received in the DRM receiver, is sent to the context manager. The context manager pre-filters the content of the XML file according to Carlos' location and sends it to the ontology together with the rest of the information. So, given his location, role, preferences, the fact that Carlos wishes to receive the information in Spanish and time constraints, the system recommends through reasoning the fast food restaurant located at the end of the Park where Carlos has been located by the GPS sensors incorporated in his mobile device.

4. *Scenario 4*: Vegetarian Worker looking for lunch.

   Carlos is a researcher attending several meetings at the San Sebastian Technology Park. It is 1 pm and he has just finished a meeting and has another one scheduled for 2 pm. He is not familiar with the area and he has no much time for lunch.

   Carlos is using a mobile device that is connected to the Technology Park digital broadcasting service, where he had his last meeting. Because he is attending a meeting, the system determines that he is an *"AttendingMeetingVisitor"*. Carlos has specified in his preferences that he likes fast-food and that he is vegetarian. So, given his location, role, preferences, the fact that Carlos wishes to receive the information in Spanish and time constraints, the system recommends through reasoning the vegetarian fast food restaurant located at the end of the Park where Carlos has been located by the GPS sensors incorporated in his mobile device.

5. *Scenario 5*: Family lunch at the Museum restaurant.

   Carlos is visiting the Science Museum located at the Technology Park with his family.

   The system reads in Carlos' agenda a scheduled visit to the museum and provides him therefore the (*"HolidayLeisureVisitor"*) role. He has gone for a walk with his family and then visited the museum. It is lunch time after the visit has finished. In addition, Carlos wishes to receive all of the information in English.

   Carlos would like to have lunch outside, but it is raining that day in San Sebastian (weather information is also received through broadcasting). The Museum restaurant offers a day-menu service, as well as some traditional *pintxos* inside, i.e. it is a covered restaurant. It is not too expensive either.

The context manager receives the XML file that has been broadcasted, Carlos' information and location. The context manager pre-processes the XML file according to Carlos' current location and sends all of the information to the ontology. Although Carlos wishes to have lunch outside in the terrace, because it is raining that day in San Sebastian, the system finally recommends him and his family the indoor museum cafeteria restaurant.

6. *Scenario 6*: Looking for a traditional restaurant.

   Carlos is attending a tourism and technology fair at the San Sebastian Technology Park. The fair takes place in one of the buildings at the main entrance of the Park.

   Carlos has never been to San Sebastian before and he enjoys tasting local traditional cuisine wherever he goes. The fair finishes at lunch time.

   The context manager gathers the XML file that is broadcasted together with Carlos' location and his personal information such as Role (*"AttendingFairRole"*) and preferences. Carlos wishes to receive the information in Spanish. The context manager pre-processes the XML file according to Carlos' location and sends all the information to the context ontology.

   Thus, given his location, role and preferences and the fact that the traditional restaurant has a medium price range, the system recommends Carlos the Basque Tavern restaurant.

7. *Scenario 7*: Day-menu restaurant at the main entrance of the Park.

   Carlos is attending a tourism technology fair at the San Sebastian Technology Park. It is taking place in one of the buildings located at the entrance of the park. Carlos is attending sessions at the fair and one of them finishes at 1 pm and the next one begins at 2 pm.

   The system determines that Carlos is an *"AttendingFairVisitor"* and through the information gathered from the agenda, the system determines that Carlos does not have too much time for lunch. The application's context manager receives information about Carlos' location, about Carlos himself and the radio broadcasting XML file. Carlos wishes to receive the information in English.

   The context manager filters this information by location and sends it to the ontology, which determines through reasoning, that the restaurant that best suits Carlos' needs at that moment is Serval.

6.2 Prototyping

As shown in Section 5.4, the CONCERT-based application for tourism provides a tourism information visualization service for visitors to easily find their way and be assisted while on the move. In addition, the interface also features the functionality to provide personal information for context manually. This information has been defined by the UNWTO[UNW08a] and has also been considered in order to build the context network of ontologies, as specified in Chapter 4, and corresponds with the data that provide information about the visitors' demographics, such as:

- *Language preference*: the language the visitor prefers.
- *Motivation*: the reason why the visitor is travelling. It can be either business or personal motivation, contemplating different alternatives for each of the options.
- *Role*: This refers to the activity visitors are involved in at a certain moment of time. For example, a particular visitor may be in a certain location due to business obligations and at the same time, the visitor may have some free time to enjoy during his journey, therefore, his motivation would be *business*, but his role would be *leisure*. This represents a subtle change in the context of visitors, that needs to be taken into account, for the potential service offered could (and possibly should) be radically different.
- *Food preferences*: the kind of food that the visitor would rather eat.
- *Agenda*: The visitors' agenda can either be read automatically from the mobile device's agenda or be manually introduced into the application. This information is then used to contemplate time constraints and the activity in which the visitor may be participating.

The laboratory prototype that has been developed for experimentation provides a simplified functionality to that explained in Section 5.4.1. The main objective of this experimentation is not to test a particular feature of the application, but rather to technically measure the system's behaviour on the one hand and to assess the extent to which people are ready and are willing to experience and enjoy these kind of services (understood as abstract concepts) on the other. In other words, to explore the potential of such contextual computing services in general and of the current application in particular.

The prototype processes in its context manager the following context information:

- *Visitor parameters*: These data has been introduced by the users themselves through the CONCERT framework-based application interface. This interface, as shown in figure 5.15, implements a panel that enables users to provide their personal information.

- *Location information*: The prototype provides the possibility to either be connected to a GPS device via Bluetooth, or to manually introduce a pretended logical location for laboratory experimentation purposes. This has been the case for both the technical and user evaluation.

- *Tourism and context information*: The prototype receives the information from the laboratory installed Content Server emulator, which at the same time becomes the information from the CIC RSS Maker (Section 5.4.1). The prototype also implements a Web service to gather weather information from the Internet, depending on the location that has been provided by the user.

This information is received in the context manager in form of an XML file with a particular schema, as explained in the previous Chapter. Then, the context manager introduces the content stored in certain tags of the XML file, i.e. keywords, in their corresponding class within the network of ontologies in form of instances. The context manager loads the ontology together with its instances into the working factory and executes the reasoning process.

Following, the context manager first checks the consistency of the network of ontologies. The consistency check consists primarily of evaluating if the ontology is formally correctly constructed, e.g. whether an instance corresponds to two disjoint classes and to check other similar potential failures. Then, the reasoning engine runs the rules to perform the rule-based reasoning. When the rules are executed, they check the variables corresponding to the specific classes that have been specified in the body of the rule. The rule has two parts, i.e. the part on the left hand side of the arrow is the antecedent and the part on the right hand side of the arrow is the consequent. If it happens that all the constituents (i.e., values of instances in the corresponding classes of the ontology) of the antecedent are true, then the consequent is executed. The following example shows one of the rules that has been implemented for the purposes of these experiments:

```
(?v rdf:type dcl:Visitor),
(?v dcl:hasMotivation dcl:AttendingMeetingMotivation),
(?v dcl:hasPreferences dcl:English),
(?r rdf:type dcl:ModernRestaurant),
```

```
(?r dcl:hasTerrace "false"^^xs:boolean),
(?r dcl:offersServiceInLanguage dcl:English),
(?r dcl:hasPrice ?p), lessThan(?p,12),
(?m rdf:type dcl:Activity), (?v dcl:isDoing ?m),
(?e rdf:type time:TemporalEntity),
(?m dcl:hasTemporalEntity ?e),
(?f rdf:type time:Instant), (?e time:hasEnd ?f),
(?f time:inXSDDateTime ?z), (?v dcl:hasScheduled ?n),
(?n dcl:hasTemporalEntity ?u),
(?u time:hasBeginning ?b),
(?b time:inXSDDateTime ?y), beforeNow(?y) ->
(?r dcl:isTourismServiceOfferedToVisitor ?v)
```

The antecedent of the rule, i.e. all of the information before the arrow, are the requirements that need to be checked by the reasoning engine. Thus, the reasoning engine checks whether there is a variable *'v'* of type *Visitor* that has the next characteristics: a certain motivation, which is *AttendingMeetingMotivation* and has English as preferred language. In addition, the rule verifies whether there is a restaurant *'r'*, which is of type *Modern* and that does not have a terrace, serves in English and has a price lower than EUR 12. Finally, the rules need to check whether the activity *'a'* that the visitor is doing at that certain moment has already finished or not. If and only if ALL these parameters have proven to be true by the reasoning engine, then, the visitor *'v'* is suggested to go to restaurant *'r'*.

6.3 Technical Evaluation

The objective of the technical evaluation presented in this section is to study and analyse the performance of the CONCERT framework-based application according to different critical variables. After a number of experiments, completed in different machines, conclusions are needed to be drawn on the performance of the framework and future work is expected to be derived from these observations and evidence found. It is necessary to evaluate the scalability of the CONCERT framework for applications where the amount of memory required for reasoning is not a restriction as well as a starting point to reduce the size of the ontology for less powerful devices, such as Smartphones. The importance of these kind of evaluations is not merely to analyse current performance, but also, to be able to predict the system's performance in future potential scenarios[Coh95].

Some arguments against experimentation maintain that there are too many variables to look at and the results would then necessarily be meaningless because the effects are swamped by noise[Tic98]. However, at least in the case of the CONCERT framework, there are just a few variables that need to be taken into consideration in order to experience and learn about its functioning.

It is important to recall again at this point, that the CONCERT framework-based application only works on regular laptops (or computers) due to the lack of existence of DRM receiver-adaptors for mobile telephones. This has allowed the prototype to use a relatively large ontology taking advantage of their reasoning potential, without specially worrying about its size and about the consequences of memory consumption on ontology rule-based reasoning. This is one crucial point of the technical experimentation for two reasons: the first one, for it allows to find out whether digital broadcasting is a valid communication technology for context-aware services in tourism. And second, if it is really a valid technology, for it permits to use observation on experiments to find the way to design very "small" ontologies. As stated in Section 4.6, the network of ontologies consists of 12 ontologies. All together there are 86 classes, 41 object properties, 22 datatype properties and 43 restrictions.

The performance experiments have been carried out populating the network of ontologies with a more or less constant number of instances, and running each time a larger number of increasingly more complicated rules on it. The minimum number of rules has been one, with one single antecedent, and the maximum number of rules used has been 8, with a maximum of 24 antecedents. Experience shows that the more number of instances within a class, the more time it takes the ontology to perform the established reasoning and the more time it takes to query it[WZGP04][GPZ94]. Therefore, the CONCERT context manager, as explained earlier in Section 5.1, always checks the existence of instances in the classes of the ontology before introducing a new one. Consequently, the number of instances has not been considered critical and the evaluation has rather focused on the number of rules executed in each of the experiments, while maintaining the order of magnitude of the triples. The reasoning tasks performed in the ontology (apart from consistency checking) correspond to checking whether a number of statements are true in order to execute a certain command, as has been explained in Section 6.2.

There is one context reasoner on the prototype that has been utilized, i.e. description logic based ontology reasoner that first checks the consistency of the network of ontologies and second, supports the rule-based reasoning. Context reasoners are built using Jena2 Semantic Web Toolkit, which supports rule-based inference over OWL/RDF graphs, Section 5.4.3. The size of the dataset is measured in terms of the number of RDF triples, each of which represents a single S-V-O predicate. In the experiments carried out for the purpose of this research, the ontology had between 780 and 790 triples, depending on the scenario.

Comprehensive experiments have been carried out using three different types of laptops, namely a MacBook Pro, a HP and a Notebook. The characteristics of these devices are specified later in their corresponding subchapters. The Filtering Engine (Section 5.4.3) has been executed straight forwardly in the Eclipse environment, where the programme code has been edited. The jconsole[2] monitoring tool has been used in order to measure the values of each of the variables in each of the experiments. The jconsole is a JMX-compliant monitoring tool. It uses the extensive JMX[3] instrumentation of the Java virtual machine to provide information on performance and resource consumption of applications running on the Java platform.

The variables that have been considered in the evaluation in each of the experiments have been the following:

- *Number of rules*: This variable refers to the number of rules that are being simultaneously executed in the experiment. The objective is to see whether there exists some kind of correlation between the number of rules running and the amount of time it takes the system to provide the result of the reasoning process.

- *Processing time*: This variable refers to the amount of time it takes the system to produce an output. Once the experiment is set, the *Run* button is pressed, and the programme begins to count the time. It finishes counting the time when the programme stops running.

- *Compilation time*: This variable refers to the amount of time spent the just-in-time (JIT) compilation.

- *Heap memory*: The Heap Memory is the runtime data for which the Java Virtual Machine (JVM) allocates memory for all class instances and arrays. The heap may be of a fixed or variable size.

- *Number of threads*: This variable refers to the current number of live daemon threads plus non-daemon threads. In the experimentation, the

[2]http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html

[3]http://java.sun.com/j2se/1.5.0/docs/guide/jmx/tutorial/tutorialTOC.html

peak number is considered, i.e. the highest number of threads since the JVM started.

- *CPU usage*: The amount of CPU resources that are used by the process at experimentation time.

The procedure followed at the time of experimentation has been described in a so called "Technical Evaluation Check-List" document. It enumerates the steps that need to be accomplished in order to guarantee equal conditions at the time of experimentation and thus, to be able to compare the results obtained in each of the experiments. The machines were re-started each time an evaluation was performed. The only programmes that were run during the experimentation were CONCERT framework-based application and the *Jconsole*, in order to collect data about the application's performance.

Once the simulation produced the results, other programmes were used in order to annotate the results and capture images, following the rest of the steps specified in the "Technical Evaluation Check-List". The date and the time were controlled as well, in order not to misplace performance graphs with performance data.

10 experiments have been accomplished in each of the machines, varying the number of rules executed simultaneously, which in fact produce the larger memory heap. The Technical Evaluation assesses the performance and memory usage of computers when they run the CONCERT framework-based application.

6.3.1 Tests on MAC OS

The first set of 10 experiments have been conducted on a MacBook Pro machine[4]. This laptop runs an OS X 10.6.1 on Intel Core 2 Duo 2.26 GHz processor with 4 GB of RAM memory, table 6.2.

Machine	**Operating System**	**Processor**	**RAM**
MackBook Pro	OS X 10.6.1	Intel Core 2 Duo 2.26 GHz	4GB

Table 6.2: MAC OS Laptop Characteristics

[4]http://www.apple.com/macbookpro/

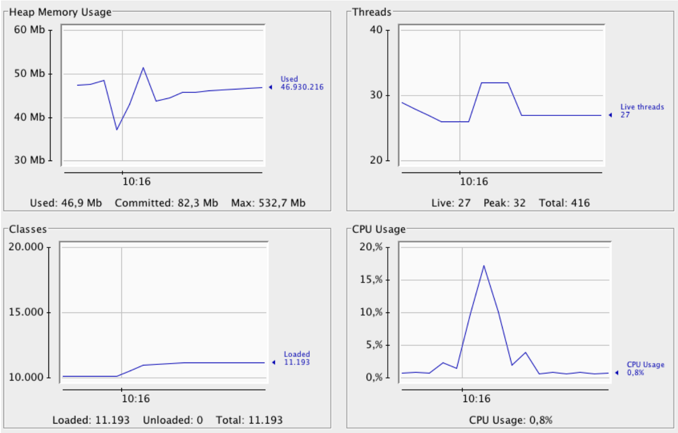

Figure 6.1: CONCERT performance in the MAC Laptop

Figure 6.1 shows the fluctuation of some of the variables (namely heap memory, number of threads, number of classes and CPU usage) according to time during one of the experiments. As shown in the figure, there is a sudden increase on the usage of heap memory at the beginning of the simulation, when the whole ontology and its rules are loaded onto the Jena model of the java programme. The reasoning consumes most of that memory and once it has been completed, there is a sudden decrease.

Table 6.3 shows the measures taken in each of the experiments carried out in the Mac OS machine.

As can be seen from the observations in table 6.3, the amount of time required by the MAC laptop to process the incoming information by means of the rules is kept under reasonable limits. It takes the system more or less between 6 to 8 seconds, 7.17 seconds on average to process the information.

The analysis of the experiments shows that the processing time is not dependent on the number of rules that are executed at the same time. The amount of heap memory on the contrary is relatively high, an average of 50.57 Mb. This is due to the reasoning process that takes place in the knowledge base when the ontology is loaded with the instances and the rules are run. A regular laptop like this one or even a less powerful one (see Section 6.3.3) can execute such a reasoning or performance, however, these values are not allowed in a mobile device, which by far could not process this reasoning

Nr. of rules	**Processing time**	**Comp. time**	**Heap Mem.**	**Threads**	**CPU Usage (%)**
1	8	2.055	53.014 MB	33	7
1	7	1.333	53 MB	32	16.5
1	7	1.326	53.316 MB	35	14.5
6	7	1.542	49.193 MB	31	11
1	7	1.376	47.253 MB	35	11,3
8	7	1.256	50.106 MB	32	25
3	7	1.359	45.264 MB	35	24.9
4	6	1.304	51.03 MB	35	16.03
3	6	1.25	45.231 MB	34	12.5
4	7	1.347	46.21 MB	33	14.3

Table 6.3: Observations on Experiments on MAC

Variable	**Observation average**
Run time av.	7.17 secs.
Heap Mem. av.	50.57 Mb
Thread Nr. av.	33
CPU av.	14.217%

Table 6.4: Average value of observations in the Mac Laptop

process. Experiments have indicated that the percentage of use of the CPU increases with the number of rules.

The number of threads and the amount of CPU usage is also more or less constant and the values of the observations are in average 33 and 14.721%. Table 6.4 summarizes the average value of the observations in the Mac experiments.

From evidence shown by the experiments, if this application would be required to be run in a real mobile phone, the concept ought to change: either the context model is hosted somewhere on the Internet, on a server for example, to be able to run the reasoning process and assume the high computational costs derived from the use of ontologies, or the size of the ontology should be decreased dramatically.

Experience and literature show, that the more number of triples in an ontology, the processing time increases exponentially[WZGP04]. Thus, in

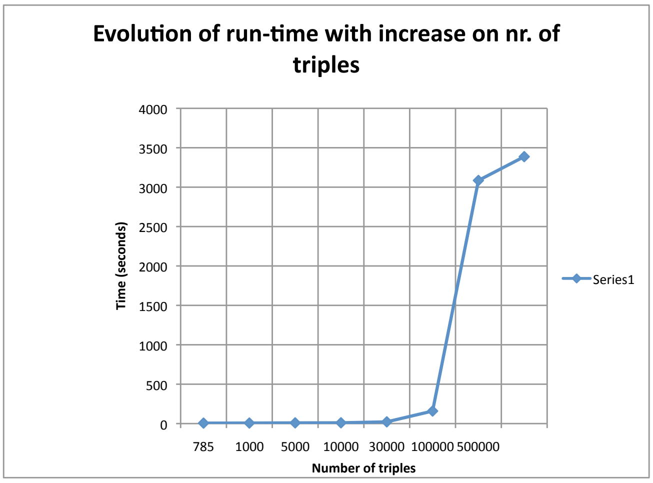

Figure 6.2: Response time vs. Number of triples on the Mac

order to verify whether this ontology also follows such a behaviour, another experiment has been carried out. The number of triples has been artificially increased and observations have been measured. The graph showed in figure 6.2 has been obtained after several observations. It shows how, in fact, the processing time increases exponentially more or less when the ontology has 30k triples.

6.3.2 Tests on Windows PC

The second set of 10 observations have been conducted on a regular HP laptop machine[5]. This laptop runs a Microsoft Windows XP Operating System, Version 2002 SP 3 on Intel Core 2 Duo at 1.86 GHz with 4 GB of RAM memory, table 6.5.

Figure 6.3 shows the fluctuation of some of the variables (namely heap memory, number of threads, number of classes and CPU usage) according to time for one of the experiments. As shown in the figure, as was the case with the Mac laptop, there is a sudden increase on the usage of heap memory at the beginning of the simulation, when the whole ontology and its rules are loaded onto the Jena model of the java programme. The reasoning consumes

[5]http://h10010.www1.hp.com/wwpc/us/en/sm/WF05a/321957-321957-64295-3740645-3955549-3782222.html

Machine	**Operating System**	**Processor**	**RAM**
Hewlett Packard	Microsoft Windows XP Operating System, Version 2002 SP 3	Intel Core 2 Duo at 1.86 GHz	4GB

Table 6.5: HP Laptop Characteristics

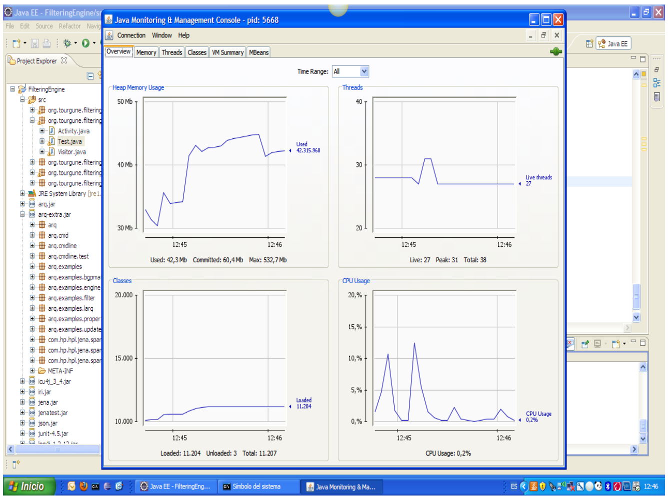

Figure 6.3: CONCERT performance in the HP Laptop

most of that memory and once it has been completed, there is a sudden decrease.

Table 6.6 shows the measures taken in each of the experiments with the HP machine.

As can be seen from the observation in table 6.6, the amount of time required by the HP laptop to process the incoming information by means of the rules is similar to the time needed by the Mac machine and despite being relatively high (an average of 7.4 seconds) is still acceptable. The meaningless increase perceived in processing time is not critical and not even noticeable by the users. Therefore, a small frequency difference of processors has no influence on the processing of the rules in the case of this network of ontologies.

The rest of the variables have provided observations that are within the expected values. It is remarkable that the amount of CPU usage is relatively

Nr. of rules	**Processing time**	**Comp. time**	**Heap Mem.**	**Threads**	**CPU Usage**
1	7	1.52	43 MB	38	12.5
1	7	1.33	42.7 MB	29	11.3
1	7	1.371	41.3 MB	29	16.2
6	8	1.322	42.5 MB	30	14.9
1	7	1.376	47.253 MB	35	11,3
8	7	1.386	43.51 MB	32	9.4
3	8	1.779	35.7 MB	32	11
4	7	1.963	38.2 MB	34	4
3	7	1.652	42.3 MB	33	11.2
4	8	1.871	39.1 MB	34	13.2

Table 6.6: Observations on Experiments on the HP Laptop

Variable	Observation Average
Run time av.	7.4 secs.
Heap Mem. av.	42.37 Mb
Thread Nr. av.	31.5
CPU av.	13.725%

Table 6.7: Average value of observations in the HP Laptop

low, a 13.725% in average, as well as in the case of the Mac machine. This amount needs to be increased in both cases, in order to make a more efficient use of resources in the CPU and possibly, lowering therefore the processing time. Table 6.7 summarizes the average results observed in the HP machine.

To finalize the experiments with the HP laptop, the number of triples of the network of ontologies has been artificially increased in order to study its behaviour, and to analyse whether it sticks to what has been observed in the previous experiment with the Mac laptop and what has been seen in the literature.

Figure 6.4 shows that the behaviour of the ontology in this case follows the expected pattern as well. However, interestingly the amount of time required to process the same number of triples in this laptop starts to increase earlier than with the Mac machine. In fact, the curve begins to have an exponential behaviour soon after the ontology has been populated with more than 10k triples. This is very likely due to the nature of the processors. Still the size of the ontology that could potentially be processed by this kind of laptop

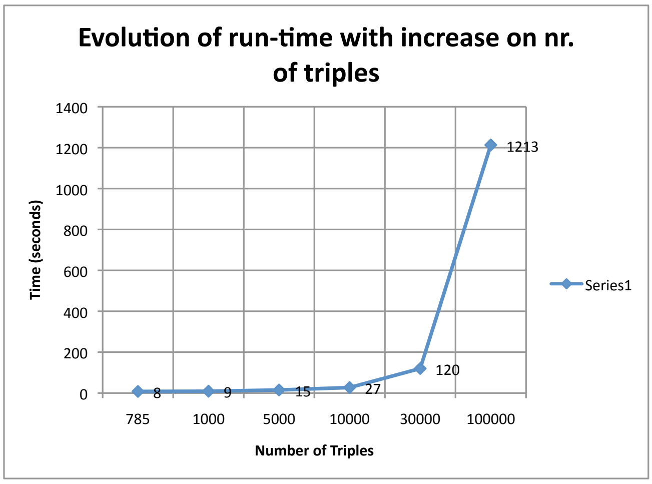

Figure 6.4: Response time vs. Number of triples on the HP

remains relatively large, and it could be considered as a machine to host a CONCERT framework-based application on the Internet.

6.3.3 Test on Windows Netbook

The third and last set of 10 experiments have been conducted on a regular Toshiba Personal Computer[6]. This laptop corresponds to one of the so called netbooks, which are becoming increasingly popular both for leisure and professional tourism, especially due to their small size, low weight and long time battery life. Besides, they are particularly well suited for Internet usage. This mini laptop runs a Microsoft Windows XP Home Edition Version 2002 SP 3 operating system on an Intel Atom processor at 1.86GHz with 1 MB of RAM memory, table 6.8.

Figure 6.5 shows the fluctuation of some of the variables (namely heap memory, number of threads, number of classes and CPU usage) according to time in one of the observations. As shown in the figure and also happens in the previous cases, there is a sudden increase on the usage of heap memory at the beginning of the simulation, when the whole ontology and its rules are loaded onto the Jena model of the java programme. The reasoning consumes

[6]http://es.computers.toshiba-europe.com/innovation/product/Toshiba-NB100-12S/1059732/toshibaShop/false/

Machine	Operating System	Processor	RAM
Toshiba Notebook	Microsoft Windows XP Operating System, Version 2002 SP 3	Intel Atom at 1.86 GHz	1GB

Table 6.8: Toshiba Notebook Laptop Characteristics

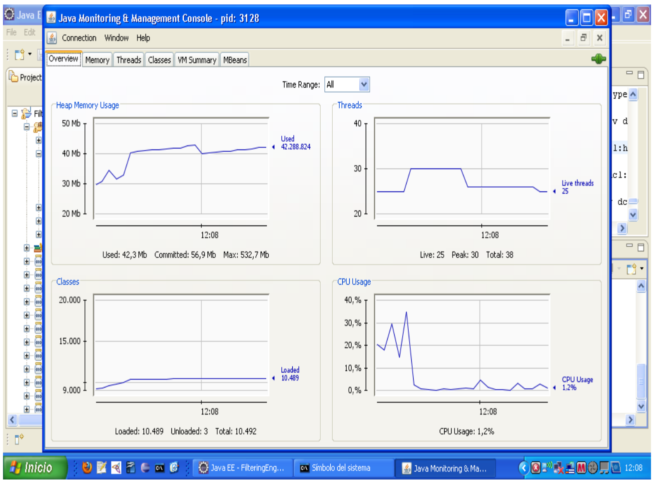

Figure 6.5: CONCERT performance in the Toshiba Laptop

most of that memory and once it has been completed, there is a sudden decrease.

Table 6.9 shows the measures taken in each of the experiments.

As can be seen from the observations, the amount of time required by this machine is more or less three times the time needed both by the Mac and the HP laptops. On average, it takes 25.8 seconds to the Toshiba netbook to process all of the information. The CONCERT framework-based application is, as said earlier, a non-time critical application, however, 25 seconds are possibly too much time for a visitor to wait to receive indications on how to move around the destination. The increase on the processing time is essentially by cause of the slower processor and also, because there is only one of them. In addition, the short amount of RAM memory considerably lowers the computer's capacity.

Nr. of rules	**Processing time**	**Comp. time**	**Heap Mem.**	**Threads**	**CPU Usage**
1	26	3.64	43 MB	29	37
1	27	3.82	40 MB	31	30
1	26	3.59	43 MB	30	35
6	25	4.12	43.5 MB	29	38
8	25	4.78	43 MB	31	32
8	25.8	3.99	42.375 MB	30	34.4
3	25	3.631	43.2 MB	30	35
4	27	3.772	42.4 MB	29	30
3	26	3.547	42.157 MB	30	31.3
4	28	3.217	43.142 MB	30	32.1

Table 6.9: Observations on Experiments on the Toshiba Laptop

Variable	Observation Average
Run time av.	25.8 secs.
Heap Mem. av.	42.375 Mb
Thread Nr. av.	30
CPU av.	34.4%

Table 6.10: Average value of observations in the Toshiba

Apart from this variable, the rest of the variables have an expected behaviour and they remain within certain ranges of acceptable values. The average of heap memory used is 42 Megabytes. This amount, despite being high, is still acceptable for this kind of device. The average thread numbers are 30 and, surprisingly the CPU usage is higher in the netbook than it is in the other two machines: in average, the Toshiba netbook makes a more efficient use of the CPU, using 34.4% in each simulation. This may have to do with RAM memory available and the nature of the processor as well. Having just one and in addition less powerful processor together with solely 1MB of RAM forces the laptop to make a very efficient use of its internal resources. Table (6.10) summarizes the results.

To finalize the experiments with the Toshiba laptop, the number of triples of the network of ontologies has been artificially increased in order to study its behaviour, and to analyse whether it sticks to what has been observed in the previous experiment with the Mac laptop and what has been seen in the literature.

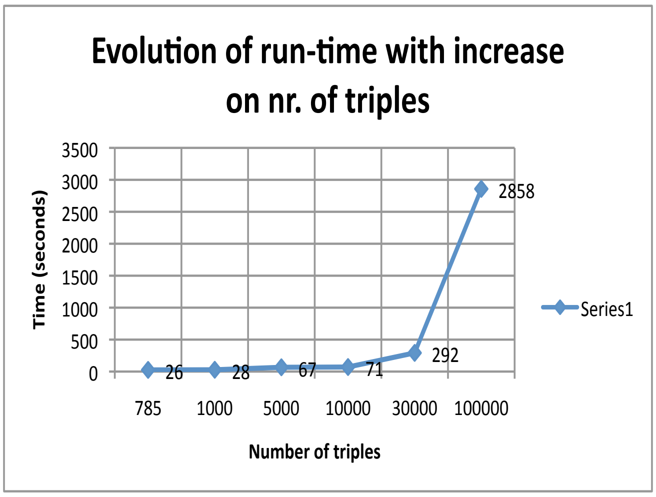

Figure 6.6: Response time vs. Number of triples on the Toshiba

The results of this experiment can be observed in figure 6.6. The machine operates as expected, however the exponential trend of the curve begins long earlier than in the previous two cases, due to the characteristics of the machine. It is remarkable that it takes about a minute with only 5k triples to process all the rules and the ontology.

6.3.4 Comparison, discussion and conclusions

Evidence of these observations show that, despite the fact that semantic technologies in general, and ontologies in particular, are a beneficial tool for contextual computing, the information processing time when it comes to run rule-based reasoning is relatively high. Both the Mac and HP laptops maintain their processing times within acceptable ranges (less than 8 seconds in both cases), however, the processing time goes up to almost 30 seconds in the case of the Toshiba laptop, which offers a CPU similar to the latest cutting edge mobile devices with 1 GHz. processors. There seems to be a tight relation between the processing time and the processor as well. Thus, as new more powerful processors come into the market, that will allow to decrease this time, a balance has to be found between the size of the ontology, instances and number of rules. An alternative to this would be to store the ontology and the rules on databases. This would very likely reduce the

Critical Variable	**Action**
Number of triples	Optimize
Processing time	Reduce
Reasoning process	Optimize
CPU usage	Optimize

Table 6.11: Summary of the actions that ought to be taken based on the observation of the technical assessment

amount of memory. However, it would penalize the processing time, which as observed already, is a critical variable on its own.

In addition, the ontology has to be refined and the reasoning process has to be made more efficient. However, the processing time is more dependant on the number of triples than on the number of rules, based on these observations and information collected from the literature. Hence, there is a need to decrease the size of the ontology, which will consequently more likely produce lower processing times and by less powerful computing capabilities.

The heap memory is more or less similar in all three machines. It is high, therefore, the size of the ontology needs to be reduced for this reason as well, so that the number of triples are decreased and thus, the whole processing time. This value being approximately similar in all three devices suggests that it is not dependant on the hardware, but on the programme itself and on the ontology. Therefore, efforts have to concentrate on working on the ontology and reducing its size in terms of number of triples and in terms of guaranteeing the lowest number of instances at the same time, as the prototype developed for the purpose of this experiments actually does.

It is surprising that the CPU usage is so low, primarily in the Mac and HP machines. The usage of the CPU increases notably in the case of the Toshiba machine, very likely due to the low power of the processor and the low size of the RAM memory. However, there is work to do in order to solve this issue, since it would on the one hand reduce the processing time and on the other, the heap memory use.

Finally, figure 6.7 shows a comparison of the behaviour of the three machines in time response, when the number of triples is increased.

Table 6.11 summarizes some of the actions that ought to be taken, based on the observations of the experiments. Nonetheless, Chapter 7 will summarize all the future work and the conclusions extracted from the research done in the framework of this dissertation.

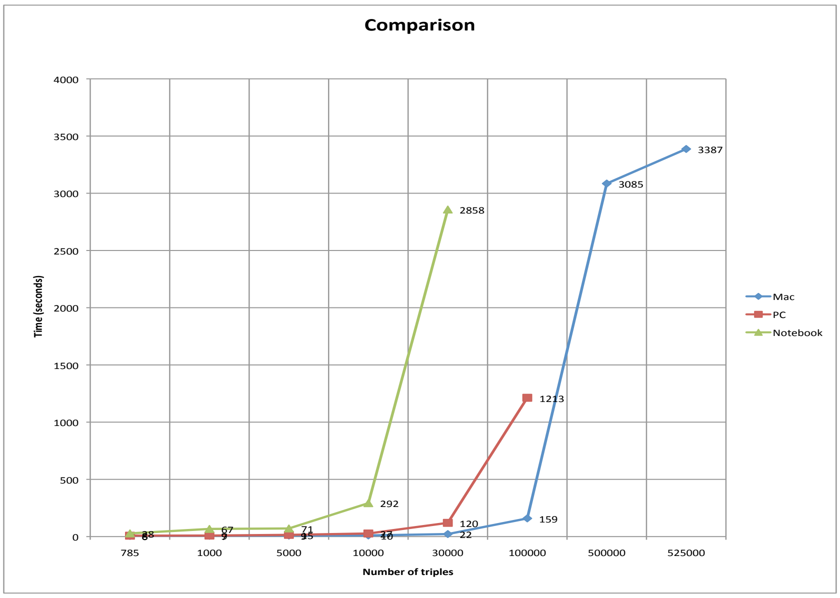

Figure 6.7: Comparison graph

6.4 User Evaluation

The final objective of the CONCERT framework is to foster the use of CONCERT framework-based contextual computing applications in the realm of tourism. Recalling now the fourth sub-hypothesis that has guided the research throughout this dissertation, i.e. *"This framework increases user satisfaction by using a semantic-based visitor-centred model for context, which is based upon recommendations provided by the UNWTO on visitor classification"*, it argues that by using an application based upon a semantic model like the one that has been built in Chapter 4 and put together in an application as shown in Chapter 5, visitors' satisfaction should be increased when *in route*. Therefore, it was concluded that it was necessary to additionally design a user-based evaluation to measure and test the initial hypothesis of the work.

Along with the increasing investment in information technologies, user technology acceptance becomes an increasingly studied topic in information systems[SZ06][Che06]. There exist a number of theories and models to assess and model user technology acceptance behaviour[VMDD03]. Most of the acceptance models have been focused on productivity-oriented or utilitarian systems, whereas recently, some work has appeared examining recreational or pleasure-oriented systems[Che06]. This being so, there are some applications that, depending on their context of use, can be utilized for reasons of productivity and support users' experience too. This could very well be the case of the CONCERT framework-based application, depending on the motivation of the journey, for example.

Despite the fact that a thorough analysis of such acceptance models is beyond the scope of this dissertation and that there is a lively discussion on them, it is important to adopt an appropriate approach in order to achieve sound conclusions based on the evidence of the observations derived from the experiments carried out with real people. Much of the previous research in this realm has used the technology acceptance model, TAM[Dav89]. TAM views user acceptance as being dependant upon the perceived usefulness (PU) and its perceived ease of use (PEOU), where PU is defined as *"the degree to which a person believes that using a particular system would enhance his or her job performance"*[Dav89], and PEOU is defined as *"the degree to which a person believes that using a particular system would be free of effort*[Dav89].

One of the most critical decisions at this point has been to identify the decision model (empirical framework) in order to have a comprehensive user evaluation, as to derive sound conclusions on whether there is any potential commercial future for a CONCERT framework-based application in the realm of tourism, should users perceive a great level of usefulness.

Summing up, in this work the main construct that has been pursued is PU. This factor has been given wide attention in the literature[Dav89] and it has shown most of the times significant values in its relation with Behavioural Intention (BI). Also, the construct PEOU has been considered appropriate to be integrated in the user evaluation empirical model, since one of the goals sought has been to identify the perceived ease of use of the CONCERT framework-based application. In addition to that, the AT (Attitude) construct has also been taken into consideration, as an indication of the tendency of users to use or not technology.

Because TAM is used as the baseline model in deriving the relations amongst the constructs in the previous model, the following hypotheses are proposed for the case of using the CONCERT framework-based application:

Hypothesis UE1: Perceived ease of use has a positive impact on behavioural intention to use the CONCERT framework-based application.

Hypothesis UE2: Perceived usefulness has a positive impact on the behavioural intention to use the CONCERT framework-based application.

Hypothesis UE3: The attitude has a positive impact on the behavioural intention to use the CONCERT framework-based application.

$$BI = f(PEOU, PU, AT, \epsilon)$$

6.4.1 User Evaluation Procedure

An experiment has been conducted on the functionality of the CONCERT framework-based application. The objective of this experiment is to test people's reaction to it and to automatically obtaining context-dependant information. In the following, the conditions in which the experiment was performed is explained.

Participants

There have been altogether 30 participants in the experiment, 56% of which were males. 40% of the participants were younger than 36 and in total 92% were younger than 50. The participants have been chosen from a random (known) group of people, trying to have as different backgrounds and competencies as possible. It is important to bear in mind that the CONCERT framework aims at being used by anyone, regardless of how skilful and used to one may be to technology. Therefore, the large variance on this variable accounts for a very diverse sample that will enhance the final results. The educational level of the sample is relatively high, since 56% of the participants had some kind of university degree. In addition, from the rest of the sample, 16% had a Ph.D. degree and 28% are on the process of obtaining such a university degree.

Despite the fact that this information has not been considered in the questionnaire, all of the participants have been tourists at some point of their lives and have been travelling on regular basis. Therefore, all of them account for a significant tourism experience and in one way or another, have always felt the need for receiving information on the move.

Material and data

The experiment was carried out with a PC laptop. It is the same one in which one set of the observations corresponding to the technical evaluation was performed. As mentioned earlier, this laptop runs on a Microsoft Windows XP Operating System, Version 2002 SP 3 on Intel Core 2 Duo at 1.86 GHz with 4 GB of RAM memory, table 6.5.

The tourism information has been created as described in Section 6.2 and uploaded to the Content Server emulator. The laboratory Content

Server has broadcasted this information to a particular IP address within the local network area at the lab where experiments have been carried out. Therefore, the IP of the laptop has been provided to the Content Server emulator and thus, the laptop has been continuously receiving tourism and context information as specified both in Section 6.2 and in Section 5.4.

In addition, participants were given a questionnaire (Appendix XXXX) they had to fill in once they had finished their experience with the CONCERT framework-based application. The questionnaire, using a four point scale, was employed to collect data for the constructs of the acceptance model proposed above. All items of the questionnaire are shown in Annex XXX. The measurements for perceived ease of use and perceived usefulness have been adapted from David's studies[Dav89, DBW92]. The measurement for behavioural intention and attitude have been adapted from another experiment performed by Chesney[Che06] and Abdalla[Abd07].

Methodology

User testing was conducted in a controlled laboratory environment. First of all, users were briefly introduced to the CONCERT framework, to the application and to the experiment's objectives. They were instructed on how to interact with the CONCERT framework-based application and before starting with the experience, they were given an example on how to manipulate the application. In addition to that, participants were given a number of scenarios, in which they had to be virtually immersed and try the different options of the application. After having become the instructions, participants were left on their own in order to accomplish the experience.

Participants were asked to put themselves into the position of a tourist and were also encouraged to recreate, as much as they possibly could, the scenarios that were given. Users had to introduce different types of personal information into the application. By changing their personal context's variables and introducing new ones, they had to experiment with the result the application would provide them with after the rule-based processing of the information by the ontology.

The experiment consisted of seven scenarios and the users had to experiment on all 7 of them. Each of them recreated different contexts or situations and therefore, ought to provide different tourism recommendations.

After having completed the user experience, each user had to fill out the above-mentioned questionnaire. On average, the whole experiment took about 15 minutes to be completed.

6.4.2 Results

Looking at the results provided by the data processing just from a descriptive point of view, it could be said that the fourth sub-hypothesis suggested at the beginning of this dissertation is clearly corroborated by the results obtained from the observation. Evidence shows that a CONCERT framework-based application, like the one that has been developed within the framework of this dissertation, improves user satisfaction.

The two most important constructs of the TAM model explained earlier, namely the PU and PEOU, have strongly been supported by participants. Table 6.12 shows the results obtained for PU.

Variable	Strong disagree	Disgree	Agree	Strong agree
PU1	0%	0%	44%	56%
PU2	0%	4%	28%	68%
PU3	0%	12%	28%	60%
PU4	0%	4%	20%	76%

Table 6.12: Results for Perceived Usefulness

Table 6.13 shows the results obtained for PEOU:

Variable	Strong disagree	Disgree	Agree	Strong agree
PEOU1	4%	20%	36%	40%
PEOU2	0%	4%	44%	52%
PEOU3	0%	12%	36%	52%
PEOU4	0%	0%	28%	72%
PEOU5	0%	0%	32%	68%
PEOU6	0%	0%	36%	64%

Table 6.13: Results for Perceived Ease of Use

For example, 56% of the participants **"strongly agrees"** on the fact that the CONCERT framework-based application supported them on the move, whereas 44% only *"agrees"* on that same matter. 68% of the participants expressed that the CONCERT framework-based application would improve their tourism experience, whereas only 4% argued that it would not have impact on it whatsoever. The application allowed 60% of the participants to more efficiently move around and 76% of the participants said that the application made it easier for them to find what they needed.

Regarding the ease of use of the application, i.e. the PEOU construct, 40% *"strong agrees"* that it is easy to provide personal information to the CONCERT framework based application, 36% *"agrees* and 20% *"disagrees.*

A Factor Analysis has been carried out in order to find other kinds of relationships amongst the variables and to confirm or refute hypotheses UE1, UE2 and UE3. Factor analysis is a method of data reduction. It does this by seeking underlying unobservable (latent) variables that are reflected in the observed variables (manifest variables). Principal components analysis has been the followed method to conduct the factor analysis.

The following table shows the total variance explained:

	Total Variance	Eigen Value	% accumulated
1	3.6	2.916	24.298
2	2.438	2.153	42.238
3	1.432	2.082	59.590
4	1.031	1.351	70.847
5	.943		
6	.614		
7	.528		
8	.416		
9	.329		
10	.280		
11	.216		
12	.171		

Table 6.14: Total Variance Explained

The determination of the number of factors to extract has been guided by theory, but also informed by running the analysis extracting different numbers of factors and seeing which number of factors yields the most interpretable results. Four factors have been identified by looking for simple structure. Eigen values are the variances of the factors and the one´s with values are over 1 have been recognized as valid. The results obtained from the variance suggest that the acceptance model may be explained by four factors. More importantly, the accumulated explanatory power of this model with base on these four factors is 70.847%, over the average explanatory power[SZ06]. In addition, the Kaiser-Meyer-Olkin Measure of Sampling Adequacy has been conducted. This measure varies between 0 and 1, and values closer to 1 are better. A value of .6 is a suggested minimum. In this case, a value of .668 has been obtained.

The type of rotations that can be done after the initial extraction of factors, including orthogonal rotations, such as varimax has been carried. The following table shows the rotated component matrix, which has determined the groups in which the used constructs and variables have been arranged.

	1	2	3	4
Using CONCERT is fun	.836			
CONCERT provides attractive UI	.817			
I like using CONCERT	.749			
Interaction with CONCERT is clear	.717			
Getting information from CONCERT is easy		.849		
Learning how to use CONCERT is easy		.738		
Using CONCERT supported me on the move		.651	.508	
Using CONCERT allowed me to move more efficiently			.903	
Using CONCERT improved my tourism experience			.847	
I had enough knowledge to use CONCERT				.775
Becoming CONCERT skilful user is easy	.463			-.583
I use technology often				.500

Table 6.15: Total Variance Explained

The table clearly shows the four groups resulting from the general linear model analysis. The groups have been given the following names: Perceived Usability, Willingness to use the system, Perceived Utility and Background Knowledge. From the results on the table, there seems to be some sort of relation between the background knowledge and the willingness to use the system.

6.5 Summary

As stated at the beginning of this Chapter, experimentation is central to the scientific process, including computer science. For this reason, this whole chapter is devoted to the (partial) evaluation of the work carried out in the framework of this dissertation. The Chapter begins in Section 6.1 by describing the different scenarios under which the CONCERT framework-based application is going to be evaluated. Then, Section 6.2 describes the prototype and the functionalities that are implemented on it for the purposes of the experiment.

Sections 6.3 and 6.4 are devoted to the technical and user evaluations respectively. The work described in this dissertation builds on a number of research questions and hypothesis introduced in Chapter 1, which are associated to the three main objectives of the dissertation:

1. To design and model a push contextual computing framework for human mobility that results in a new definition of context and that is centred on the mobile nature of people.

2. To gather contextual and tourism information from Web distributed and heterogeneous sources of information and disseminate it via digital broadcasting. No external sensors shall be used, except for the ones embedded in mobile devices. Content gathered in the Web will then be adapted for broadcast. This will constitute the first level of interoperability proposed by the new framework, i.e. interoperability at the infrastructure level.

3. To design, model and implement an ontology-based model of context, using networks of ontologies in order to increase the model's consistency, reasoning capabilities, modularity, interoperability, re-use and sharing. The ontology will particularly focus on modelling the visitor according to tourism established vocabulary and will implement a rule-based reasoning engine to filter incoming information. This will constitute the second level of interoperability proposed by the new contextual computing framework, i.e. interoperability at the model level.

The objective of the evaluations carried out in this Chapter, in relation to the three objectives is to validate the hypotheses against the experimental results obtained. As mentioned at the beginning of this Chapter, the experimental process carried out in this chapter has been used to validate sub-hypotheses 3 and 4, since the other two sub-hypotheses and the two research questions have already been supported in Chapter 5, as for having built the application itself.

The importance of experimentation in computer science has been emphasized at the beginning of the Chapter, without going to deep into that debate. Then, the scenarios and the settings for the different evaluations have been described, the rational and criteria followed in these activities explained and, finally, the results have been interpreted both in a qualitative and quantitative way.

The third and fourth sub-hypotheses are enumerated following and their coverage in terms of the results of the evaluations are shown:

- *H3*: A network of ontologies approach can be used to model visitor context and use it as a tourism information filtering engine based on description logics (DL) and rule-based reasoning.

  This sub-hypothesis is linked to the third objective and partially to the second. Throughout this dissertation, a claim has been done to support the idea that broadcasted information could be further manipulated by the use of semantic technologies once it was received in the client. The main reason for pursuing the use of digital broadcasting in this dissertation is to test whether it could potentially be used in the future as communication technology in tourism mobile services. This way, telephone communication fees under roaming conditions and having to use the contextual computing application in a restricted sensor populated area would be avoided. In addition, existing context models have shown to be insufficiently adapted to the domain of human mobility, hence not completely covering the requirements of context modelling in terms the nature of the information that needs to be taken into consideration.

  This sub-hypothesis suggests that a network of ontologies can be used to filter incoming information based on the DL reasoning capacities inherent in ontologies themselves and through a rule-based reasoning engine. In order to validate this statement, an application has been developed (showed in Section 5.4) and adapted in Section 6.2 that models context by means of a network of ontologies and that uses Jena based rules in order to filter the incoming information.

  In combination with a simple to use user interface, which provides visitors with the required options about their personal context. By means of the menus described earlier (Sections 5.4.4 and 5.4.5), visitors are able to introduce their personal characteristics into the context model that further uses this information in order to filter the incoming information. The observations of the experiments implemented with three different kinds of machines showed evidence that in fact, a network of ontologies approach can be used to model the context of a visitor.

The functionalities implemented in the filtering engine java project guarantee the following: first, the context manager checks the consistency of the context model for the given value of the context parameters and then, *"once and if and only if"* the current model is formally consistent with the original conception of the network of ontologies, the rule-based reasoning was executed and results obtained. This approach has provided evidence that indeed, incoming broadcasting information may be further manipulated by semantic technologies to reduce the amount of information to display on the screen of a mobile device. In the evaluation performed in the framework of this dissertation, the Mac OS and HP machines have performed better (less time) than the Toshiba machine.

The characteristics themselves of the kind of communication technology used (digital radio broadcasting) allows to use fairly large ontologies. However, despite the fact that ontologies are really valuable tools and that they are extensively used in ubiquitous computing, the processing time appears to be a critical factor. In two of the three machines used for the evaluations the processing time was less than 8 seconds. This is an acceptable time, for this is a non-time-critical application. However, the results obtained with the third machine are very high and possibly not acceptable by users.

- *H4*: This framework increases user satisfaction by using a semantic-based visitor-centred model for context, which is based upon recommendations provided by the UNWTO on visitor classification.

  The main objective pursued by the CONCERT framework is to make contextual computing applications universally usable and accessible. As has been stated throughout the dissertation, the use of these kinds of systems is highly restricted to laboratory environments and their use is fairly complex. In addition, the analysis of the existing context ontologies performed in the literature review, it was identified that the nature of the human *in route* was not actually considered in such ontologies, therefore, the consequent reasoning that they could perform would not produce the expected results for visitors. Chapter 1 and Chapter 2 recall, based upon evidence found in the literature review both on the realm of tourism and on the realm of context-aware research, that tourism is a *push* sector, i.e. visitors are not expected to be actively involved in the search of information that may be useful for them. It is also argued in those Chapters that the restrictions that mobile devices present for a comprehensive interaction with them, would discourage users from using context aware information if visitors were to actively search for information. Therefore, there is a need for a new approach to a context model that actually considers the information requirements that characterize the nature of human mobility.

There are different ways and information sources to accomplish this. In the case of this dissertation, the information provided by the United Nations World Tourism Organization for definition and characterization of tourists has been used, as shown in [UNW08a]. The information provided in that document has been organized in an ontology (Chapter 4) and the ontology has been incorporated into an application (Chapter 5).

The number of people participating in the evaluation of the CONCERT framework-based application (36) have shown to have widely different relations with technology, which actually enhances the results of the observation, for not only very technology-wise skilful people have participated in the sample that has been observed. This statistical observations usually require fairly large samples, however, this first approach conducted in this research work, allows to extract interesting conclusions, which should be interpreted cautiously, on the use of a contextual computing application in tourism from a quantitative point of view.

The application was used by the participants on the experiment without intervention of any other agent (human or machine) and only required some initial comments on the utilization of the tool.

Therefore, evidence was found that participants in general support these kinds of contextual computing services in tourism, through the utilization of the adequate tools, i.e. mobile devices. Additionally, evidence was also found that these kinds of systems increase the user experience.

During the evaluation processes, it was observed that while few participants would have needed real training in order to facilitate a faster adaptation and adoption of these kinds of applications, their performance was good. All participants quickly assimilated the way the CONCERT framework-based application functions and how personal information may be introduced in the application. The experimentation process was fairly quick, which allowed them to see results rather quickly and therefore, increased their encouragement.

Apart from the questionnaire that all participants in the evaluation process had to complete, there were comments on how to improve the interface and on what would an (so to say) *ideal* user interface be like. It was emphasized that the interaction with the interface was low, so that users could not feel overwhelmed by the amount of information they had to introduce.

Summing up, this work should be considered as a first step towards future research on contextual computing in tourism.

Chapter 7

Conclusions

"The show must go on... "
Freddy Mercury, (1946 - 1991), British musician.

This dissertation presents a number of contributions to the state of the art in context-awareness that aim at addressing the different research problems found in this discipline and described in Section 1.1. The final objective of the CONCERT framework is to set the path to foster a wider use of context-aware systems and applications by the general public, more specifically by tourists or visitors, i.e. people *en route*, making the final transition from laboratory environments, where context-aware systems have been developed in a reasonable successful way, to the market place. Thus, visitors would be benefited and assisted on the move by (automatically, i.e. *push*) receiving context-based relevant information for them, enhancing their tourism experience and being able to more easily find their way in an unknown environment, as demonstrated by evidence in the previous Chapter.

This scenario has not been previously feasible given the intrinsic complexity of context-aware systems and applications. As has been shown throughout the dissertation, existing approaches have important limitations due to both their infrastructure dependency and high development costs derived from the sentient-based environments they require. Besides, the use of such systems and applications is not obvious and would possibly discourage users. In addition, no deep theoretical studies exist on the nature of context in the framework of human mobility, which has created existing approaches in this realm not to be successful in terms of general use.

The contributions proposed by this dissertation address these and other limitations (as specified previously) and are deemed to make possible a more widespread adoption of contextual computing services in the realm of tourism as well as in other frameworks. This will, of course, also be boosted by the dramatic development of mobile devices and communication technologies.

Section 7.1 summarizes the most relevant contributions put forward by this dissertation. It first brings back the two research questions, the hypothesis and the objectives set in Section 1.4 and describes and analyses the extent to which they have been achieved. Of course, there have been difficulties on the way, and the decisions that had to be taken have necessarily influenced the remaining course of this dissertation. This is the reason why there is a short sub-section dedicated to discussion on these matters.

Section 7.2 illustrates a number of research and scientific implications, not only for contextual computing in tourism but also for the discipline of contextual computing in general.

Both the technical and user evaluations carried out in Chapter 6 have enabled to elucidate areas that deserve further investigation. In addition, the limitations of the CONCERT framework as well as of the application that has been developed based upon it and the limitations of the dissertation itself as a research work with a well established scope, have also based the identification of other working areas for the future. All of these new research lines are specified in Section 7.3.

The Chapter, and therefore the dissertation, close in Section 7.4 with some general remarks and comments.

7.1 Contributions

Section 1.1 describes the problems found in context-aware research and some of the motivations that have activated this dissertation. These have led to set the research framework for this piece of work in terms of research questions, the hypothesis against which the research work has been validated and finally, the associated goals of this dissertation. The analysis of the literature has added additional elements to the existing ones and has thus completed the research agenda.

Hence, the background presented in Chapter 1 and Chapter 2 have led to the theoretic work presented in Chapter 3. This theoretic work has then been divided in practical terms into two kinds of developments:

- First, the development of the context model based on the theory, explained in Chapter 4.
- Second, the development of a system architecture and an application in Chapter 5 that also incorporate the model of context implemented in Chapter 4.

The validation of the research work has been accomplished continuously. In first place, there is a validation of the context model at the end of Chapter 4. The network of ontologies has been evaluated on its own, independently from the rest of the architecture before all the components of the prototype have been put together. Secondly, the research questions and the first two sub-hypotheses have been validated in Chapter 5 by developing the CONCERT-enabled applications and by making it work in the scenarios that have been described in Chapter 6. Finally, empirical observations have been carried out in Chapter 6 as to find out how the application operates and to analyse the extent to which humans are ready to use such an application. Evidence obtained from these observations allows to confirm the validity of the second two sub-hypotheses.

So, it is now time to recall these questions and the hypothesis that have been put forward at the beginning of the dissertation, together with their associated goals and to describe the degree to which they have been covered by the contributions.

The general research questions that have guided the piece of work proposed in this dissertation are:

- *How can contextual computing information based applications be implemented with minimal infrastructure and deployment overload?*
- *Can ontologies be used to filter incoming push broadcasted information according to the values of context at a given location and time?*

In order to provide an answer to the general research questions, the following hypothesis, which has been further subdivided for clarity in four sub-hypotheses, has been put forward:

- *H(1)*: A push contextual computing framework can be established by gathering context information from Internet information sources without external sensors.
- *H(2)*: Digital broadcasting can be used to disseminate context and tourism information to mobile devices equipped with a filtering engine designed to support human mobility.
- *H(3)*: A network of ontologies approach can be used to model visitor context and use it as a tourism information filtering engine based on description logics (DL) and rule-based reasoning.
- *H(4)*: This framework increases user satisfaction by using a semantic-based visitor-centred model for context, which is based upon recommendations provided by the UNWTO on visitor classification.

These research questions and sub-hypotheses led to a number of general goals, which are specified next:

- *GOAL 1*: To design and model a push contextual computing framework for human mobility that results in a new definition of context and that is centred on the mobile nature of people. (Section 3.1.2).

- *GOAL 2*: To gather contextual and tourism information from Web-distributed and heterogeneous sources of information and disseminate it via digital broadcasting. No external sensors shall be used, other than those embedded in mobile devices. Content gathered in the Web will then be adapted for broadcast. This will constitute the first level of interoperability proposed by the new framework, i.e. interoperability at the infrastructure level. (Chapter 3).

- *GOAL 3*: To design, model and implement an ontology-based model of context, using networks of ontologies in order to increase the models consistency, reasoning capabilities, modularity, interoperability, re-use and sharing. The ontology will particularly focus on modelling the visitor according to tourism-established vocabulary and will implement a rule-based reasoning engine to filter incoming information. This will constitute the second level of interoperability proposed by the new contextual computing framework, i.e. interoperability at the model level. (Chapter 4).

The literature review in the realm of context-awareness shows that (i) neither does consensus exist on a definition for the notion of context nor do existing ones suit the requirements to support human mobility, (ii) a sufficiently agreed model for context management does not exist, (iii) current works reveal the need for a scientific approach to study context as a discipline on its own right and within the realm of tourism and finally, (iv) the use of sensors to gather contextual information poses serious limitations and pre-requisites for making contextual-computing systems universally utilized.

The following list enumerates the contributions of this dissertation and their relation to the different objectives, hypothesis and general research questions.

1. *New theoretic approach to the notion of context*: As has been mentioned earlier in the dissertation, context-awareness has been approached from several distinct research disciplines. The evolution of mobile devices and communication technologies have allowed to use these kinds of systems in tourism, in form of tourism mobile guides. However, there

has never been a deep study of the characteristics of the notion of context in such a framework, i.e. human mobility. In fact, as mentioned earlier, the notion of context in human mobility has directly been imported from the human computer interaction research discipline. This weak theoretic support is possibly one of the reasons why context-aware services are not yet more present in real life. The theory development presented in Chapter 3 aims at providing the necessary theoretic support to the notion of context and context-awareness in the realm of human mobility. It first studies the nature and characteristics of people on the move and then, presents a theory of context based on those assumptions.

2. *New definition to the notion of context*: The review of the literature shows a number of definitions of the notion of context. Most of them have been suggested from the human computer interaction community. However, the idea of the concept of context is so vast that it is practically impossible to set a definition of context that may be valid for all and any of the research disciplines. Context is contextual and therefore, its application in different domains will very much depend on the characteristics of such domains. The theoretic development presented in Chapter 3 has lead to a new definition of the notion of context that is valid in the realm of human mobility, i.e. tourism. The objective is not to provide yet another definition of context. Rather, to provide an integrative and operative definition, that is valid for the framework of human mobility.

3. *A new pragmatic approach to the notion of context*: Traditional context-aware systems take into consideration a vast number of variables in order to very precisely determine the value of context of particular individuals at a given moment of time. The large amount of such variables considered adds necessarily various difficulties in terms of sensor infrastructure, communication protocols, maintenance tasks, etc. Besides, despite the fact that these variables, e.g. level of noise, level of lightning, amongst others, provide context information, they happen not to be really relevant. In addition, the more variables considered in the model of context, the more number of triples the ontology comprehends and therefore, the more computing resources are needed when it comes to perform reasoning tasks.

   Thus, the pragmatic approach taken in the conception of this dissertation allows to:

- First, reduce the implementation load to develop a context-aware framework and application (related to Research Question 1).
- Second, it allows to dramatically reduce the infrastructural costs of such applications, since, the theoretic approach presented earlier does not consider the use of sensors to gather contextual information. In addition, this application also allows to use these (sensor free) kinds of applications anytime, anywhere, since no restrictions are posed by the existence of sentient environments (related to Goal 2)

4. *New context information requirements*: The definition of context put forward in Chapter 3 is divided into various levels. Each of these levels indicates the kind of information that is required to represent and manage it. Therefore, this piece of information would constitute a specific information requirement to make the context model useful in operative terms. Therefore, despite the fact that there are already some "established" requirements in terms of context information according to the literature (specified in Chapter 2), the new definition of context has several implications in this regard, and new information requirements have been identified and set for the case of human mobility, as has been specified in Chapter 3. (related to Goal 1).

5. *The two level interoperability framework*: As mentioned in Chapter 2, there are two trends on context-aware research. The applications developed by the first trend had severe pre-conditions and limitations for their use outside their frameworks, derived from their theoretic conception and from the technological development. In fact, the context management modules of those architectures could function only with the specified module of context information gathering. The second trend tried to build interoperable context-aware applications by use of ontologies to model context, however, their applications were still too dependant on their particularities and that did not allow to scale or re-use them outside the environment of their conception. In addition, the use of sensors restricted the area in which these applications could fully exploit their functionalities.

   Tourism, understood in its wider way as human mobility, is basically an outdoor environment, i.e. it happens anytime and anywhere. The single fact of suggesting to populate large portions of cities, regions or countries with sensors to gather contextual information is just not affordable. Context information ought to be possibly obtained in an anywhere, anytime manner as well, as for it not to impose restrictions in terms of where or where not to use these applications. This is the reason why this dissertation presents the new concept of double level of interoperability:

- The first level of interoperability comes from the fact that not using sensors allows to use other sources of context information, which are more flexible (related to Goal 2).
- The second level of interoperability comes from the fact of using networks of ontologies to model context, enhancing the level of interoperability of existing context ontologies (related to Goal 3).

6. *Experimenting with digital broadcasting*: technology and nature. At the beginning of context-aware research, the applications were used in indoor environments because there was no powerful wireless technology that enabled to disseminate information to long distances. The development of communication technologies have brought new wireless technologies, e.g. 3G, 4G, HSDPA, GPRS, Wi-Fi, WiMAX just to mention a few, that allow people to be connected to sources of information in anytime, anyway manners. These technologies being extremely useful to allow mobile connections, are exceptionally expensive as well under certain circumstances, and hardly any attention has been given to this fact. Access to information ought not to be limited because of the pricing of telecomm operators, it ought to be more open. In addition, the aforementioned communication technologies are real *connection* technologies, i.e. people have to actively establish a connection and interact with mobile devices to access the piece of information that they require. Tourism is more the other way around, i.e. ideally (relevant) information ought to be (automatically) served to people on the move. This is the underlying idea of *push* based technologies, an example of which is digital broadcasting.

   Digital broadcasting has been successfully used in a number of sectors for information transmission. However, it has never been used in the realm of tourism, despite tourism being an adequate sector to use this technology, and as far as the literature review is concerned, there has been no experimentation of digital broadcasting with semantic technologies. This has been one of the research questions in fact. And the way this has been faced and developed in the framework of this dissertation demonstrate that in reality, digital broadcasting technologies may be used not only for tourism information

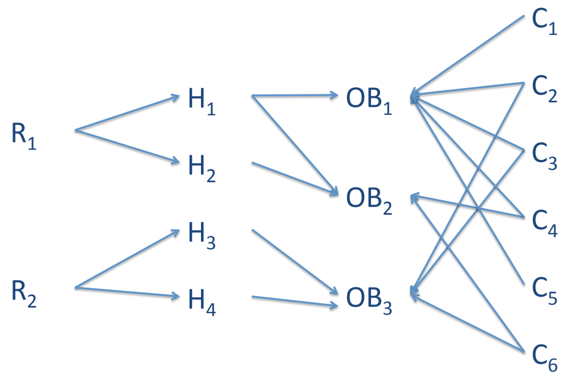

Figure 7.1: Graphical representation of the impact of the contributions

transmission, but also, they can be semantically manipulated to increase user satisfaction, as hypothesized in Section 1.4.

Tables 7.1 and 7.2 summarize the contributions put forward by the CONCERT framework and their relations with the objectives, hypothesis and research questions. The first column shows the relevant issues found in the related literature. The second column shows the gaps and problems existing approaches have. Finally, the last column shows the proposal of this dissertation to cover the gaps.

Figure 7.1 represents graphically, the relation there is among the research questions (R), the hypothesis (H), the general goals (OB) and the contributions (C).

Related Work	Existing Gaps and Problems	Contribution
• Theory of Context	• Notion of context not attempted from the human mobility perspective. • Not sufficiently agreed definition of context.	• Context in Human Mobility, (Sect. 3.1). • New definition of context (Sect. 3.1.2). • Focus on visitors' context rather than applications' context.
Models of context: • Not common context model. • Non-semantic models[Wan92, SAW94, DA00a]. • Semantic models[SLP04, CFJ04b, GPZ94, WZGP04, PdBW+06].	• No common ontologies nor usage of ontologies. • No re-usable models. • Ad-hoc models.	• ContOlogy: focus on visitors (Sect 4.5). • No-constraints model: continuously adapting to changes in context

Table 7.1: Summary of theoretic contributions of this dissertation (I).

Related Work	**Existing Gaps and Problems**	**Contribution**
Context gathering: • Networks of sensors	• Systems not ubiquitous. • Systems dependant on location of sensors. • Added complexity. • Many prerequisites for applications to run.	• Two level interoperability. Internet as context source (Ch.3). • Only mobile embedded sensors. • Fewer prerequisites to make systems work under more general conditions.
Connectivity Technologies: • Wi-Fi, 3G, UMTS, GPRS.	• Connectivity is expensive. • Even more under roaming conditions.	• Digital broadcasting (Sect.5.2).
Location of Systems: • Indoor or in restricted sentient environments.	• Tourism is typically an outdoor environment.	• An application that runs outdoor (Sect.5.4).

Table 7.2: Summary of theoretic contributions of this dissertation (II).

7.1.1 Discussion

As mentioned in the introductory part of this Chapter, there have been a number of decisions that had to be taken in the course of the current work. In addition, there are some limitations that originate from the fact of using one or another technology, derived from the application scenarios, the type of data, etc. All of these, both the decisions that had to be taken and the delimitations, have directly influenced the directions the work has taken. At the end of the day, this work is what it is, due in a great extent to such factors and it is worth to briefly discuss about them, as a token of self-criticism necessary in any piece of research.

The following issues are considered critical in the development of this piece of research.

Limitations of Digital Broadcasting

As mentioned several times throughout this dissertation, there are various technologies available to provide context-aware applications. However, since the very beginning of this piece of research, the interest and the efforts are put in discovering whether digital broadcasting can be a useful technology in the realm of tourism for the purpose of contextual computing services. The motivation comes not only from its characteristics, i.e. very well suited for the *push* based niches and (low) cost, but because other sectors have already been testing and working with digital broadcasting services.

The objective was to test whether digital broadcasting technologies can be used to provide contextual computing services in tourism through semantic rule-based processing. Evidence obtained from the research and development work in the framework of this dissertation has corroborated this fact, i.e. digital broadcasting can be used for these purposes.

Nonetheless there are a number of limiting factors that need to be taken into account:

- *Reception of Digital Broadcasting*: The standards mentioned in Chapter 5, i.e. DAB, DRM and DRM + are not yet available for mobile phones, Smartphones, PDAs and such devices. The reason for this is that these kinds of devices are not yet equipped with an appropriate receptor and therefore, they would not be able to distinguish the information signal in this frequency. Through and *ad hoc* receiver-adaptor, this signal is available in regular laptops, like the ones that have been used for the technical evaluation in the framework of this dissertation.

- Data transfer: The amount and the type of information that can be broadcasted is also limited. This technology is not like any other real connection technology. It can broadcast limited amounts of information and in given types.

- *Communication*: There is not DRM mobile transmitter yet, i.e. digital broadcasting can be received in mobile devices but mobile devices are not yet prepared to send any information by means of this technology. For the purposes of the experiments developed in the framework of this dissertation, this is not relevant, however, if digital broadcasting contextual computing services were to be implemented, this fact ought to be taken into consideration. *Push* based services are necessary in tourism, but in addition to that, visitors may want to pose queries, or (actively) look for information themselves. For this reason, until there is such a broadcasting technology available, how digital broadcasting can work together with other technologies is something interesting researchers ought to look at.

Tourism modelling and Tourism categorization

This is one of the crucial parts of the model, i.e. tourist categorization or classification on the one hand, and modelling of tourism objects on the other.

There is a vast amount of theory dedicated to user modelling, both in and outside the realm of tourism. There are even lots of ontologies dedicated just to user modelling. There are also different approaches to the modelling of tourism objects based on different techniques. Since the CONCERT framework focuses on people on the move not necessarily for leisure purposes, it has followed the classification provided by the UNWTO for visitors. This classification, as considered in the context ontology, contemplates the diverse motivations visitors could have in order to undertake a certain journey. Other tourist classification ontologies focus, for example, on providing the right recommendation for sightseeing opportunities, and are not so appropriate to model the framework of mobility itself in a more wider and general way.

On the same way, there has been a significant effort in the last years dedicated to build a tourism ("*standard*") ontology. As well as in other sectors, e.g. in the realm of cultural heritage with the case of the CIDOC CRM standard ontology (ISO 21127:2006), the tourism domain does not have a standard ontology. In spite of that, there are many examples of tourism ontologies. For the purposes of this dissertation and the experiments carried out in this research, a regular classification of tourism objects has been followed and no research efforts have been invested in this task, since it is beyond the scope of this dissertation.

User evaluation Constructs

User evaluation is one of the most important parts of the whole evaluation process performed in this dissertation. In fact, one of the sub-hypothesis reflects upon the increase of user satisfaction that an application such as the CONCERT framework-based application developed in this dissertation would produce in visitors.

There are various technology acceptance models in the literature. The TAM model has been chosen despite its limitations and the fact that it is widely used in productive environments, rather than in leisure environments. However, the prototype that has been used in the user evaluation is not a final product. The evaluation performed was just a preliminary test to find out the extent to which people would be ready to experience these kinds of services.

The TAM model considers two important factors, i.e. PU and PEOU. These have been regarded crucial within the evaluation in order to find out from observation whether there is or not evidence of support to the application. This is the reason why the TAM has been used.

Context history

Context History represents the information about past values of context of a particular individual that has been stored in a repository. This is a really valuable source of information, since it can help predict future values of context given one current situation, for example.

Context history is an element that has actually been taken into account, as can be seen in the CONCERT architecture depicted in figure 5.1. However, the information stored in the context history repository has not been considered for reasoning processes. The objective was to find out whether digital broadcasted information could be used by semantic technologies, and the development of the framework and the CONCERT framework-based application have stick to that research question leaving aside other issues, despite the fact that they can be valuable and interesting. Context history will be included in future versions of its application.

Security, privacy and trust

Security, privacy and trust are matters that are increasingly being investigated in the last years in all kinds of information systems. Even more so considering the generalized use humans do of the Internet for almost anything: restaurant bookings, bank transfers, purchasing, amongst others. Security and privacy of personal data is a must in any information system and also, in mobile services, such as the one presented in this dissertation.

User security, privacy and trust is in fact something many researchers in the realm of context-awareness have paid attention to[Wan92, CFJ04b, DA00a]. This is a critical issue since context aware systems and applications need to make use of very sensitive personal information in order to deploy their full functionality, e.g. user identification, information about demographics, location, amongst others.

The CONCERT architecture presented in Chapter 5 considers a security, privacy and trust module that takes care of such an important issue. However, and again due to the same reasons stated for the case of Context History, the functionality of this module of the architecture has not been taken into account in the system that has been developed in this dissertation.

Having said this, however, the use of digital broadcasting implicitly increases user security and privacy. In fact, the client, i.e. the mobile device, receives all of the information and the model of context is actually hosted in the client. This way, all the personal information that users provide into the application remain in the mobile phone and are not uploaded or sent to other servers or computing machines to compute that information. The computation is done locally, therefore, guaranteeing security and privacy and intuitively increasing trust of this kind of systems.

Sources of context information

One of the most critical issues faced during this dissertation has been to decide the sources of context information. Experience has shown that existing approaches are too dependant on sentient environments, i.e. areas that have been populated with different kinds of sensors in order to gather ambient information. However, these systems have turned out to be extremely dependent on these sentient environments, and their use has been highly restricted by this fact. In addition, tourism happens anytime anywhere, therefore, new sources of context information had to be sought. The Internet is one of the best information sources ever and, even if there are kinds of information that will not be considered in the CONCERT model, the Internet has been chosen as the source for context information.

The model suggested in this work presents however a number of limitations. As conventional sensors are proposed not to be used, the applications' contextual information has to rely on Web-based information, i.e. data is assumed to be accurate and that it is being continuously updated.

Furthermore, there are some kinds of data that can only be obtained with sensors, e.g. noise level, lightning level, as mentioned earlier. Therefore context information in the CONCERT framework is not as rich as it could by the use of these kinds of sensors. However, it is worth mentioning that Smartphones in the near future will be equipped with more and more diverse type of sensors, such as accelerometers, thermometer and so on, contributing to an enrichment of context information. This information however is believed not to be strictly relevant for a tourism application and furthermore, the fact of not having sensors makes it easier to make Contextual Computing applications universal. The trends indicate that there will be more and more sensor data available on the Web and the future Internet of Things will enable to set dialogues among various mobile devices, allowing context information exchange between them.

7.2 Research and scientific implications

It is mentioned in the summary of Chapter 2 that context is contextual. This is indeed one of the most important scientific and research implications provided by this research work. The notion of context, as can be seen in this dissertation is sufficiently wide and comprehends too many variables and constituents as to be attempted from one single point of view, or from a single research discipline. In addition, the notion of context in the framework of human mobility has enough entity and particularities as to be studied on its own. Hence, the approach to the notion of context needs to necessarily be different. It is not about a completely new scientific discipline, it is about changing the vision and looking at things on a different way. Previous work is of course (partly) valid, but that said, there are new characteristics that need to be taken into consideration.

The new theory implied by this new approach has implications as well in the kind of information that needs to be taken into account when creating the model of context. This information has to be accessed in new ways, and there may be practical implications in this regard as well, as specified in Chapter 3.

There are also important implications in terms of having to find new ways, new challenges to elaborate methods or techniques that actively search for context and tourism information on the Web, both in formal and informal information repositories.

The need to involve real people on the process is also essential. The objective of the systems and applications that may derive from the CONCERT framework is to serve people in an easy way. The computational cost of ontology-based applications is very high, therefore a good balance has to be found between the computing costs and what is actually computed. A deep social research is needed with real travellers in order to determine the kind of information they expect.

7.3 Publicactions

During the development of this dissertation, feedback has been obtained in various ways, as indicated by the research methodology presented in Section 1.5. In addition, significant effort has been dedicated to disseminate the partial results to different conferences and workshops.

The following list presents the most relevant publications to the dissertation written or participated by the author:

- **Carlos Lamsfus**, Aurkene Alzua-Sorzabal, David Martín and Diego López de Ipiña "Ontology-based Situational Information Rule Filtering to Support Human Mobility" In Review for the *2010 ACM Conference on Ubiquitous Computing.*

- **Carlos Lamsfus**, Aurkene Alzua-Sorzabal, David Martín and Diego López de Ipiña "Digital Broadcasting for context-aware services in tourism". In Print for the *2nd Region 8 IEEE Conference on the History of Telecommunications* to be held in Madrid on 3-5 November 2010.

- **Carlos Lamsfus**, Christoph Grün, Aurkene Alzua-Sorzabal and Hannes Werthner. "Context-based matchmaking to enhance tourists' experience". *European Journal for the Informatics Professional, CEPIS-Upgrade Journal*, Number 203. Pp. 17 - 23. 2010.

- **Carlos Lamsfus**, Aurkene Alzua-Sorzabal, David Martin and Zigor Salvador. "Semantic-based Contextual Computing support for Human Mobility", *In Information and Communication Technologies in Tourism 2010*, Eds: Ulrike Gretzel, Rob Law and Matthias Fuchs. Springer Verlag, pp. 603 - 615. ISBN-978-3-211-99406-1. ENTER Conference, Lugano Switzerland, 2010.

- **Carlos Lamsfus** and Aurkene Alzua-Sorzabal. "Computación Contextual basada en Semántica en el ámbito de la movilidad humana" *CIC Network*, pp. 50 – 54

- **Carlos Lamsfus**, Aurkene Alzua-Sorzabal, David Martín, Zigor Salvador and Alex Usandizaga. "Human-Centric Ontology-Based

Context Modelling in Tourism," *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, KEOD 2009, pp. 424 - 434. Madeira, Portugal, October 2009

- **Carlos Lamsfus**, Aurkene Alzua-Sorzabal, David Martín, Zigor Salvador and Alex Usandizaga. "Contextual Computing Based Services in Tourism," *Proceedings of the 4th Mediterranean Conference on Information Systems*, MCIS 2009, pp. 677 - 690. Athens, Greece, September 2009

- Alejandro Cadenas, Carlos Ruiz, Iker Larizgoitia, Raúl García-Castro, **Carlos Lamsfus**, David Martín, Iñaki Vázquez, Marta González, María Poveda. "Context Management in Mobile Environments: a Semantic Approach", *In proceedings of the 1st Workshop on Context, Information and ontologies of the 6th European Semantic Web Conference*, CIAO Workshop at the ESWC, 2009. Iraklion, Greece, June 2009.

The following, presents another list of publications. They are not directly linked with the topic of this dissertation, however, it is worth mentioning them here as a representation of the research trajectory. One relevant publication out of this list is the paper published in the Doctoral Symposium of the International Semantic Web Conference in 2005:

- Aurkene Alzua-Sorzabal, **Carlos Lamsfus**. "Measuring Competitiveness: ICT as an innovation tool in Tourism". *In Proceedings of the European Chapter TTRA Conference.* Rotterdam, The Netherlands. April 2009

- María Teresa Linaza, Frederik Löhöffel, Ander García, **Carlos Lamsfus**, Aurkene Alzua, Ainhoa Lazkano. "Mash-up Applications for Small Destination Management Organizations Websites" *In Information and Communication Technologies in Tourism 2008*, pp. 130-140. ISBN 978-3-211-77279-9. Innsbruck, Austria.

- María Teresa Linaza, Ander García, Ana Susperregui, **Carlos Lamsfus**. "Interactive Mobile Assistants for Added Value Cultural Contents. *VAST*. 2006

- Aitor Urbieta, Carlos Gómez, Guillermo Barrutieta, Marta González, Joseba Abaitua, JosuKa Díaz, **Carlos Lamsfus**. "Semtek: An advanced research project about agents' theory applied to context semantic adaptation." *IADIS International Conference on applied Computing 2006.* Donostia - San Sebastián, Spain.

- **Carlos Lamsfus**, María Teresa Linaza, Tim Smithers. "Re-use and alignment of ontologies: the art-E-fact ontology as an extension of the CIDOC CRM". *In proceedings of the Doctoral Symposium of the ISWC*, pp. 5 - 9. November 2005, Galway, Ireland.

- **Carlos Lamsfus**, María Teresa Linaza, Tim Smithers. "Towards semantic-based information exchange and integration standards: the art-E-fact ontology as a possible extension to the CIDOC CRM (ISO/CD 21127) standard". *In Proceedings of the Workshop on Integrating Ontologies, of the K-CAP conference*, pp. 49 - 54. (ISSN 1613-0073. Banff, Alberta, Canada. October, 2005.

- Ander García, **Carlos Lamsfus**. "Digital Security Training Laboratory". *IADAT, -tcn, 2005.* Portsmouth, England. September, 2005

- Ander García, **Carlos Lamsfus**. "Virtual Tutors and Graphical Spatial Metaphors to support an eLearning platform in Digital Security Training". *IADAT-e2005.* Biarritz, France, July 2005.

- Gorka Marcos, Héctor Eskudero, **Carlos Lamsfus**, María Teresa Linaza. "Semantic Based data Retrieval from a Cultural Knowledge Database". *4th International Workshop on Content Based Multimedia Indexing.* CBMI, 2005. Riga, June 2005.

- Gorka Marcos, Héctor Eskudero, **Carlos Lamsfus**, María Teresa Linaza. "Data Retrieval from a Cultural Knowledge Database" *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005).* Montreux, Switzerland, April 2005.

- **Carlos Lamsfus**, Iñigo Barandiarán, Ander García, Jorge Posada, Joxe Mari Elola, Igor Unanue, Jesús Colet. "An eLearning platform to support vocational training centres on digital security training". *Training, Education and Simulation International, TESI 2005.* Maastricht, The Netherlands. March 2005.

- **Carlos Lamsfus**, Georgios Karagiannis, Sophia Sotiropoulou, Héctor Eskudero, Gorka Marcos, María Teresa Linaza Linaza, sister Daniilia. "The art-E-fact ontology: a possible contribution to the CIDOC CRM". *CRM and 10th CIDOC CRM SIG Workshop.* Nürnberg, Germany. December, 2004.

- María Teresa Linaza, Héctor Eskudero, **Carlos Lamsfus**, Gorka Marcos. "An Authoring Tool For Digital Storytelling". *The 5th*

International Symposium on Virtual Reality, Archaeology and Cultural Heritage, VAST 2004. Brussels, Belgium. December, 2004.

- María Teresa Linaza, Héctor Eskudero, **Carlos Lamsfus**, Gorka Marcos and Julián Flórez. "Authoring Digital Storytelling in Mixed Reality: the art-E-fact Project". *1º Workshop Luso-Galaico de Artes Digitais*. ARTECH 2004. Lisboa, Portugal. July, 2004.

7.4 Future work

Despite the fact that this dissertation addresses a number of open issues in context-awareness, it has to necessarily set its scope and has therefore some limitations. In addition, evidence shown by experimentation has raised up new questions that deserve special research efforts.

Following, there is an enumeration of the open issues that need to be addressed:

1. *Research on context theory*: Real tourists have to be implicated in defining the notion of context, mostly, when it comes to refine the idea of context in order to minimize its corresponding computing model, so that it can be hosted in a mobile device and processed there. Surveys to real visitors ought to allow to elucidate the kind of information they would expect from such a system. This would of course have its implications in the original theory of context put forward in this dissertation and it ought to be changed according to the observations.

2. *Research on ontology refinement*: The great amount of computing and processing resources needed to compute an ontology and its reasoning process is due to the number of triples of the ontology. If an ontology is to be used in a mobile device of the type of a Smartphone, a lot of work has to be done on the refinement of the ontology, i.e. how to make the ontology smaller in terms of number of triples so that it can be computed in a mobile device.

   Real visitors shall be involved in an experimental phase of the ontology development as well. This will allow to find out more about intentions and motivations of visitors *en route* in order to include them into the network of ontologies. This is something that has not yet been considered in other context ontologies and as has previously been stated, the participation of real users may improve the usefulness of the final (networks of) ontologies[vDHS07].

3. *Research on reasoning optimization*: Ontologies are very valuable computing tools, for the functionality they have to reason on both implicit information and on rule-based reasoning. However, as stated earlier, it is also well known that the reasoning process has high computing costs (see Section 6.3). This sometimes makes researchers reluctant when it comes to use ontologies in their computing applications. The experiments run to find out the operational reality of the CONCERT framework-based application has indicated that the memory consumption in the reasoning process is very high. This, as well as for the number of triples is due to the requirements of the reasoning process itself. In order to alleviate the computing costs of the reasoning process, the following issues could be taken into account in the future:

    - *Distribution of the reasoning process*: reasoning is usually performed in one single machine, be it the server or the client. Context-aware traditional approaches have usually performed all the reasoning tasks in a server in order to optimize battery lives in mobile devices and to take advantage of computing power of servers with respect to mobile devices. But the question is: could something be done to run part of the reasoning on the server and then, the other part of the reasoning on the client?
    - *Implementation of the reasoning process*: rule-based reasoning is based on the rules that have to be written in advance for a particular programme. There is no limit on the number of rules that can be used in a particular application. On the contrary, the number of rules ought to be as high as possible, since that way, all the potential cases could be considered and therefore, a better service offered. In this sense, is there any way to generate rules automatically? Can learning algorithms be applied to learn rules, and to derive new rules with base in what they have learnt?

4. *The Future Internet*: There is big debate now on what is the future's Internet is going to be like. It has various names, such as Internet of Things, Internet of Services, the Future Internet. Anyway, given that the Internet is the main source of both context and tourism information for the CONCERT framework, since CONCERT is about using the Internet of and for people, major changes in this infrastructure may have crucial implications for the future development of CONCERT framework-based applications in terms of content and knowledge. This may have implications in the theory of context, in the own nature of

mobile phones, and so on. The impact the evolution of the Internet may have is called in this work Context of Things. It is essential to understand how the Future Internet is going to impact on Context models, even more considering that the presented new paradigm does not consider to use conventional sensors to gather contextual information.

Linked somehow with the notion of the Future Internet, cloud computing enables all services to be accessed through the Internet. This new concept is key for the future and will bring about massive social change. People will no longer have to rely on a connection to their office network since cloud computing will allow them to work and access all resources and services from any location and through any device.

5. *Communication Technologies*: Further research is needed on connectivity technologies. The existing ones provide mobile internet access to a reasonable cost as long as the service is not provided under roaming conditions, which considerably raises the connection price. Wi-Fi, RFID, Bluetooth and other connectivity technologies could help on the way. More research is necessary as well on middleware technologies and platforms to find out the extent to which they can support Contextual Computing applications' efficiency in a domain-divided world.

6. *Contextual computing and middleware*: The spread of mobile devices in these kinds of architectures (context-aware architectures) makes it necessary to consider some extra issues regarding this mobile world. Surprisingly, the physical limitation of these devices are not an impediment on themselves, but because of the additional issues that have to be considered, e.g. distribution, scalability, modularity, mobility, privacy, failure tolerance and even battery life and network connections. Due to these aspects, many existent platforms choose to develop middleware for context-aware systems.

7. *Crawlers and meta-search engines to enrich contextual information*: The digital broadcasting architecture showed in Chapter 5 gathers information from the Internet. How this happens and the development of crawlers that perform this functionality is beyond the scope of this dissertation. In fact, the CONCERT framework considers that it has all kinds of necessary contents available. However, in an advanced version of this system it could be interesting to have crawlers and

meta-search engines that continuously look for updated (context and tourism) information on the Internet so that this information can enrich the service provided to visitors.

8. *Recommender systems in the future enriched with Web-based contextual information*: In line with the previous future research line, recommender systems in the future, could also use Web-based contextual information as input so that they produce a better result.

9. *Ontologies and efficiency of ontology use*: reflexive ontologies. A reflexive ontology is *"A reflexive ontology is a description of the concepts, and the relations of such concepts in a specific domain, enhanced by an explicit self-contained set of queries over the instances"*[TSSP08]. A reflexive ontology or self contained ontology increases the performance of the ontology when the ontology is queried. How this concept could help to reduce the processing time of the network of ontologies with respect to other critical variables, such as memory consumption or CPU usage is still an open issue. It would be worth to investigate this fact to decide whether to incorporate it or not in future versions of the application.

10. *Digital Radio Broadcasting*: the use of digital radio as a new communication technology. As mentioned earlier in this Chapter, how digital broadcasting technology can work with other available communication technologies at the same time is something that still is a matter of research. Since digital broadcasting allows only to receive information, other technologies ought to be used if the user wants to query other sources of information. For example, using Wi-Fi, RFID or Bluetooth as uplink communication channel could be tested. Therefore, how to combine the use of digital broadcasting for incoming information and the use of any traditional connectivity technology for other purposes has to be undertaken in future research, as long as there is no digital broadcasting based uplink channel.

11. *Uncertainty in contextual computing reasoning*: When humans reason over certain facts, the decision taken is most of the time given a certain weight, i.e. a probabilistic value. There is a degree if uncertainty in the reasoning process that has not been considered in the model of CONCERT implemented for the purposes of this dissertation. However, being able to measure the probability with which a certain situation may happen is something interesting in contextual computing

and that future iterations of the CONCERT framework ought to look at.

12. *Multilinguality*: Languages are an integral and often defining part of cultures. Until around the turn of the millennium the treatment of multilingual data in computer systems posed major problems. Any Semantic Web-based technologies approach ought to provide access to information in several languages, allowing the creation and access to SW content independently of the local language of content for their purposes. This is even so more relevant in the case of tourism, where issues of cultural independency have to be guaranteed, whereby users anywhere anytime may access information that is relevant to them in their preferred languages. New techniques must be explored that differ from the usual hypertext structure visualization of the current web[BCCGP02].

    Linked to multilinguality, one of the trends for the future is semantic-based automatic translation. Automatic translation will remove language barriers by enabling people to speak and understand other languages. This will lead to a huge expansion in the amount of information available and to greater reception capacity while visitors are on the move.

13. *Self-management contextual computing applications*: Self managing contextual computing systems, such as service discovery that leverage a warning in the mobile device depending on the context. This would be something similar as to having the mobile device in sleeping mode, and having it automatically activated by a service that is offered within the context of the visitor, precisely, because of the value of such visitor's context[ABB+09].

7.5 Final remarks

If someone was to ask about the status of context-aware research, given that there are not publicly available solutions yet, the inquiry be something like *"is contextual computing research dead?"*. The answer to this question would have to be double. On the one hand, the answer could very well be "no", since as a matter of fact, researchers in various disciplines continue to build such context-aware applications, as demonstrated by the literature review. However, on the other hand, the answer could be "yes", since no substantive theoretic advance is being done and still these systems and applications only live in laboratories due in a great extent, to a weak theoretic support.

There are still enormous opportunities to make significant contributions to the core body of knowledge of this discipline starting from the current status. Thus, a number of ambitious challenges have been set and faced at the beginning of this dissertation. A new theoretic approach to the notion of context has been built. Then, based in the previous, a new model and architecture has been proposed, in order to foster the use of these kinds of systems and applications by everyday users. The evolution of mobile devices and their proliferation in society, the advancement of connectivity technologies and the trend towards creating hybrid spaces (symbiosis between nature and technology) will trigger a radical change in the way persons involved in mobility, or 'visitors' in operative terms, interact with their environment.

From the results provided by the work (theoretical, practical and experimental) done in this dissertation, the future challenges foreseen, discussions, other ongoing research work, the agendas of technological platforms (both in Spain and in Europe), and the general trends, it is expected that there will be new ways in which humans will interact with their (hybrid physic-digital) environment, not only indoor, but also outdoor environments.

The results of this dissertation are expected to have a significant impact on contextual computing in general and on contextual computing in tourism in particular, that boost the relationship of humans with their ever more digital environment. Therefore, context-based applications will not only shape the future of tourism services, but will also provide with the opportunity to better understand human behaviour in the digital society of the future.

However, it is not longer only valid to develop prototypes, applications or systems. Computer Science has to be given a higher degree of *"scientificness"* by experimenting more with the prototypes developed and by involving humans on the loop. Paraphrasing Tichy, *"[...] Without experiments Computer Science is in danger of drying up and becoming an auxiliary discipline. [...]"*.

With the theory developed in the framework of this dissertation and its consequent prototype, I hope to have minimally contributed to the status of core knowledge of contextual computing and to set the basis for consistent future research and scientific activities based on CONCERT.

Appendix I

First Appendix: Ontology Requirement Specification Document (ORSD)

This Appendix shows the Ontology Requirements Specification Document of the ContOlogy network of ontologies derived from the specification process presented in Chapter 4.

I.a First section

Ontology requirements specification document

The following tables show an example of an Ontology Requirements Specification Document (ORSD) with a summary of all the information that has been determined through the different tasks presented in Section 4.5.2.

<table>
<tr><td colspan="2">ContOlogy Context Modelling Requirement Specification Document</td></tr>
<tr><td>1</td><td>Purpose</td></tr>
<tr><td></td><td>The purpose of building the ContOlogy network of ontologies is to provide:
An adequate framework for context management in the realm of tourism.
Consensus on the way context is modelled and managed in the realm of tourism.
An adequate framework to provide contextual based services in tourism.
A modular conception of the notion of context that derives in:
A set of modular and scalable ontologies that are defined in the OWL DL standard language[W3C04b].
An easily extensible network of ontologies that represent other domains of interest within the travel and tourism domain that are also expressed in OWL DL.
A step forward to make contextual computing services in tourism available to the general public in an efficient and sustainable manner.</td></tr>
<tr><td>2</td><td>Scope</td></tr>
<tr><td></td><td>The development of ContOlogy is motivated by the need to build a new context model based on the requirements and theoretic contributions presented in Chapter 3. The lack of consensus both on the definition of the notion of context and on establishing a context modelling and management method have originated a number of different (non sufficiently established) context models and definitions. Therefore, in order to address these limitations, a new framework for contextual computing in the realm of tourism has been suggested. In practical terms, this has implied a new definition of context, as well as a new set of context information modelling and management requirements.</td></tr>
<tr><td>3</td><td>Level of formality</td></tr>
<tr><td></td><td>The ContOlogy network of ontologies has been developed in the OWL DL language</td></tr>
</table>

Table I.1: ContOlogy ORSD (I)

4	Intended Users

	The intended users of this ontology are the following:
	
	• *User 1*: Visitors. They shall be the main users of this ontology. They will not manipulate the ontology themselves, but they ought to experience its functionality through the CONCERT-based application (Section 5.4)
	
	• *User 2*: Tourism authorities and destination management organizations (DMOs). They can use this ontology to:
	  – Provide content and publish tourism related information.
	  – Measure and analyze tourists flows in order to support knowledge-based decision making processes.
	  – Do marketing, e.g. DMOs or tourism companies can advertise themselves.
	
	• *User 3*: Travel and tourism industries: to market their products in the Tourism Objects ontology (Section 4.6), so that they can be offered to visitors.

5	The intended uses of the ontology are the following:
	• *Use 1*: To model the context of people on the move considering characteristics typically related to the framework of human mobility. • *Use 2*: To support semantic categorization of visitors. • *Use 3*: To boost the use of semantic technologies to represent tourism related information and services. Tourism authorities and destination management organizations (DMO) may be interested in marketing their products in this ontology. • *Use 4*: To foster a standard semantic representation of tourism objects. • *Use 5*: To foster a *de facto* standard semantic model of context. • *Use 6*: To provide tourism information based services on the move. • *Use 7*: To support search of tourism objects.

Table I.2: ContOlogy ORSD (II)

<table>
<tr><td>6</td><td>Groups of Competency Questions</td></tr>
<tr><td></td><td>The groups of competency questions are the following:
Visitor (19 CQ).
Device (6 CQ).
Network (8 CQ).
Location (8 CQ).
Weather (6 CQ).
Time (6CQ).
Tourism Environment / activity (9 CQ).</td></tr>
<tr><td>7</td><td>Pre-glosary of terms</td></tr>
<tr><td></td><td>Terms</td></tr>
<tr><td></td><td>Visitor (Freq. = 6).
User (Freq. = 2).
Language (Freq. = 2).
Device (Freq. = 9).
Network (Freq. = 8).
Location (Freq. = 5).
Weather (Freq. = 1).
Time (Freq. = 5).
Tourism Environment / activity (Freq. = 9).</td></tr>
</table>

Table I.3: ContOlogy ORSD (III)

Appendix II

Second Appendix: The ContOlogy Network of Ontologies

This appendix shows the OWL file containing the ContOlogy network of ontologies (See Chapter 4).

II.a OWL File of ContOlogy

The following OWL code is only part of the ContOlogy network of ontologies. The rest of the file has been omitted for clarity.

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY time "http://www.w3.org/2006/time#" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!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#" >
    <!ENTITY DefinedClasses "http://www.owl-
    ontologies.com
```

```
    /DefinedClasses.owl#" >
]>


<rdf:RDF xmlns="http://www.owl-ontologies.com
/DefinedClasses.owl#"
     xml:base="http://www.owl-ontologies.com
     /DefinedClasses.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:time="http://www.w3.org/2006/time#"
     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:DefinedClasses=
     "http://www.owl-ontologies.com
     /DefinedClasses.owl#">
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource=
        "http://www.w3.org/2006/time"/>
    </owl:Ontology>

    <!--
    ////////////////////////////////////////
    //
    // Object Properties
    //
    ////////////////////////////////////////
     -->

    <!-- http://www.owl-ontologies.com/
    DefinedClasses.owl#hasEnvironment -->

    <owl:ObjectProperty rdf:about="#hasEnvironment">
        <rdf:type rdf:resource=
        "&owl;TransitiveProperty"/>
        <rdfs:domain rdf:resource="#Environment"/>
        <rdfs:range rdf:resource="#Environment"/>
    </owl:ObjectProperty>
```

```
<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasMotivation -->

<owl:ObjectProperty rdf:about="#hasMotivation">
    <rdfs:range rdf:resource="#Motivation"/>
    <rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasPreferences -->

<owl:ObjectProperty rdf:about="#hasPreferences">
    <rdfs:range rdf:resource="#Preferences"/>
    <rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasRole -->

<owl:ObjectProperty rdf:about="#hasRole">
    <rdf:type rdf:resource=
    "&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="#Role"/>
    <rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasScheduled -->

<owl:ObjectProperty rdf:about="#hasScheduled">
    <rdfs:range rdf:resource="#Activity"/>
    <rdfs:domain rdf:resource="#Visitor"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasSkyCondition -->

<owl:ObjectProperty rdf:about=
"#hasSkyCondition">
```

```
    <rdfs:range rdf:resource="#SkyCondition"/>
    <rdfs:domain rdf:resource=
    "#WeatherConditions"/>
</owl:ObjectProperty>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasTemporalEntity -->

<owl:ObjectProperty rdf:about="#hasTemporalEntity">
    <rdfs:range rdf:resource=
    "&time;TemporalEntity"/>
</owl:ObjectProperty>


<!--
//////////////////////////////////////////////
//
// Data properties
//
/////////////////////////////////////////////
 -->

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasAge -->

<owl:DatatypeProperty rdf:about="#hasAge">
    <rdfs:domain rdf:resource="#Visitor"/>
    <rdfs:range rdf:resource="&xsd;decimal"/>
</owl:DatatypeProperty>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasPrice -->

<owl:DatatypeProperty rdf:about="#hasPrice">
    <rdfs:domain rdf:resource=
    "#TourismRestaurantServices"/>
    <rdfs:range rdf:resource="&xsd;int"/>
```

```
</owl:DatatypeProperty>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#hasRestaurantName -->

<owl:DatatypeProperty rdf:about=
"#hasRestaurantName">
    <rdfs:domain rdf:resource=
    "#TourismRestaurantServices"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<!-- http://www.w3.org/2006/time#inXSDDateTime -->

<owl:DatatypeProperty rdf:about=
"&time;inXSDDateTime"/>

<!--
///////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////
 -->

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Ability -->

<owl:Class rdf:about="#Ability">
    <rdfs:subClassOf rdf:resource="#Preferences"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Activity -->

<owl:Class rdf:about="#Activity">
    <owl:disjointWith rdf:resource="#Device"/>
    <owl:disjointWith rdf:resource=
    "#Environment"/>
```

```
    <owl:disjointWith rdf:resource=
    "#KindOfTourismConcepts"/>
    <owl:disjointWith rdf:resource="#Location"/>
    <owl:disjointWith rdf:resource="#Motivation"/>
    <owl:disjointWith rdf:resource="#Network"/>
    <owl:disjointWith rdf:resource="#Preferences"/>
    <owl:disjointWith rdf:resource="#Role"/>
    <owl:disjointWith rdf:resource="#Visitor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#AgeRangeDemographics -->

<owl:Class rdf:about="#AgeRangeDemographics">
    <rdfs:subClassOf rdf:resource=
    "#Demographics"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Area -->

<owl:Class rdf:about="#Area">
    <rdfs:subClassOf rdf:resource="#Environment"/>
</owl:Class>

<!-- http://www.owl-ontologies.com
/DefinedClasses.owl
#BusinessMotivation -->

<owl:Class rdf:about="#BusinessMotivation">
    <owl:equivalentClass>
        <owl:Class>
            <owl:oneOf rdf:parseType="Collection">
                <rdf:Description rdf:about=
                "#AttendingCongressMotivation"/>
                <rdf:Description rdf:about=
                "#AttendingMeetingMotivation"/>
                <rdf:Description rdf:about=
                "#AttendingConferenceMotivation"/>
                <rdf:Description rdf:about=
                "#AttendingFairMotivation"/>
```

```
            </owl:oneOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Motivation"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#City -->

<owl:Class rdf:about="#City">
    <rdfs:subClassOf rdf:resource="#Environment"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#CoveredRestaurant -->

<owl:Class rdf:about="#CoveredRestaurant">
    <rdfs:subClassOf rdf:resource=
    "#TourismRestaurantServices"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Demographics -->

<owl:Class rdf:about="#Demographics">
    <rdfs:subClassOf rdf:resource="#Preferences"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Device -->

<owl:Class rdf:about="#Device">
    <owl:disjointWith rdf:resource=
    "#Environment"/>
    <owl:disjointWith rdf:resource=
    "#KindOfTourismConcepts"/>
    <owl:disjointWith rdf:resource="#Location"/>
    <owl:disjointWith rdf:resource="#Motivation"/>
    <owl:disjointWith rdf:resource="#Network"/>
    <owl:disjointWith rdf:resource=
    "#Preferences"/>
```

```
        <owl:disjointWith rdf:resource="#Role"/>
        <owl:disjointWith rdf:resource="#Visitor"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/
    DefinedClasses.owl#DeviceHardwarePlatform -->

    <owl:Class rdf:about="#DeviceHardwarePlatform">
        <rdfs:subClassOf rdf:resource=
        "#DeviceTerminalHardware"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/
    DefinedClasses.owl#DeviceSoftwarePlatform -->

    <owl:Class rdf:about="#DeviceSoftwarePlatform">
        <rdfs:subClassOf rdf:resource=
        "#DeviceTerminalSoftware"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/
    DefinedClasses.owl#DeviceTerminalBrowser -->

    <owl:Class rdf:about="#DeviceTerminalBrowser">
        <rdfs:subClassOf rdf:resource="#Device"/>
        <owl:disjointWith rdf:resource=
        "#DeviceTerminalHardware"/>
        <owl:disjointWith rdf:resource=
        "#DeviceTerminalSoftware"/>
        <owl:disjointWith rdf:resource=
        "#DeviceTerminalType"/>
    </owl:Class>


    <!--
    ///////////////////////////////////////////
    //
    // Individuals
    //
    //////////////////////////////////////////
     -->
```

```
<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Arbelaitz1 -->

<owl:Thing rdf:about="#Arbelaitz1">
    <rdf:type rdf:resource="#ModernRestaurant"/>
    <hasPrice rdf:datatype="&xsd;int">30</hasPrice>
    <hasTerrace rdf:datatype="&xsd;boolean"
    <false</hasTerrace>
    <offersServiceInLanguage rdf:resource=
    "#English"/>
    <isLocatedAt rdf:resource=
    "#TechnologyParkCentralArea"/>
</owl:Thing>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#Arbelaitz2 -->

<ModernRestaurant rdf:about="#Arbelaitz2">
    <rdf:type rdf:resource="&owl;Thing"/>
    <hasPrice rdf:datatype="&xsd;int">22</hasPrice>
    <hasTerrace rdf:datatype=
    "&xsd;boolean">false</hasTerrace>
    <offersServiceInLanguage rdf:resource=
    "#English"/>
    <isLocatedAt rdf:resource=
    "#TechnologyParkCentralArea"/>
</ModernRestaurant>



<!-- http://www.owl-ontologies.com/
DefinedClasses.owl
#AttendingConference -->

<Activity rdf:about="#AttendingConference">
    <rdf:type rdf:resource="&owl;Thing"/>
    <hasTemporalEntity rdf:resource="#Finish"/>
</Activity>
```

```
<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#AttendingConferenceMotivation -->

<owl:Thing rdf:about="#AttendingConferenceMotivation">
    <rdf:type rdf:resource="#BusinessMotivation"/>
</owl:Thing>

<!-- http://www.owl-ontologies.com/
DefinedClasses.owl#AttendingConferenceVisitor -->

<Role rdf:about="#AttendingConferenceVisitor">
    <rdf:type rdf:resource="&owl;Thing"/>
</Role>


<!--
/////////////////////////////////////////////
//
// General axioms
//
/////////////////////////////////////////////
 -->

<rdf:Description>
    <rdf:type rdf:resource="&owl;AllDifferent"/>
    <owl:distinctMembers rdf:parseType=
    "Collection">
        <rdf:Description rdf:about=
        "#AttendingCongressMotivation"/>
        <rdf:Description rdf:about=
        "#AttendingMeetingMotivation"/>
        <rdf:Description rdf:about=
        "#AttendingConferenceMotivation"/>
        <rdf:Description rdf:about=
        "#AttendingFairMotivation"/>
    </owl:distinctMembers>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&owl;AllDifferent"/>
    <owl:distinctMembers rdf:parseType=
```

```
        "Collection">
            <rdf:Description rdf:about=
            "#ReligionPilgrimageMotivation"/>
            <rdf:Description rdf:about=
            "#HolidaysLeisureMotivation"/>
            <rdf:Description rdf:about=
            "#EducationalTrainingMotivation"/>
            <rdf:Description rdf:about=
            "#HealthMedicalMotivation"/>
            <rdf:Description rdf:about=
            "#VisitingFriendsRelativesMotivation"/>
            <rdf:Description rdf:about=
            "#ShoppingMotivation"/>
            <rdf:Description rdf:about=
            "#TransitMotivation"/>
            <rdf:Description rdf:about=
            "#OtherMotivation"/>
        </owl:distinctMembers>
    </rdf:Description>
    <rdf:Description>
        <rdf:type rdf:resource="&owl;AllDifferent"/>
        <owl:distinctMembers rdf:parseType="Collection">
            <rdf:Description rdf:about=
            "#HealthMedicalVisitor"/>
            <rdf:Description rdf:about=
            "#VisitingFriendsRelativeVisitor"/>
            <rdf:Description rdf:about=
            "#ShoppingVisitor"/>
            <rdf:Description rdf:about=
            "#AttendingFairVisitor"/>
            <rdf:Description rdf:about=
            "#AttendingConferenceVisitor"/>
            <rdf:Description rdf:about=
            "#HolidayLeisureVisitor"/>
            <rdf:Description rdf:about=
            "#AttendingMeetingVisitor"/>
            <rdf:Description rdf:about=
            "#ReligionPilgrimageVisitor"/>
            <rdf:Description rdf:about=
            "#AttendingCongressVisitor"/>
            <rdf:Description rdf:about=
```

```
            "#TransitVisitor"/>
        </owl:distinctMembers>
    </rdf:Description>
</rdf:RDF>



<!-- Generated by the OWL API (version 2.2.1.1138)
http://owlapi.sourceforge.net -->
```

Appendix III

Third Appendix: Journaline Viewer Data Structure

This appendix shows an example of an XML file that follows the structure required (including the definitions of some keywords) in order for it to be broadcasted and subsequently displayed by the Journaline viewer.

III.a CIC RSS Content Maker File Example

The following XML code represents the file that has been used with the purpose of both the Technical and User Evaluations (see Chapter 6) in the framework of this dissertation. The XML code shows just an example of the code that can be used for this purpose.

```
<?xml version="1.0" encoding="UTF-8"?>
<journaline xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:noNamespaceSchemaLocation=
"http://www.w3.org/2001/XMLSchema-instance">
  <menu>
    <title>Welcome to the Technology Park
    Tourism Information Service!!</title>
    <message>
      <objectparameters>
        <defaultlanguage iso="spa"></defaultlanguage>
      </objectparameters>
      <title>
        Serval
        <keyword description="@offersServiceInLanguage:
        Spanish=Spanish@area=Entrance@keyword=
```

```
    @hasTerrace=false@hasPrice=12@type=Traditional">
    </keyword>
  </title>
  <body>
    <text></text>
  </body>
</message>
<message>
  <objectparameters>
    <defaultlanguage iso="eng"></defaultlanguage>
  </objectparameters>
  <title>
    Kutxa Espacio Restaurant
    <keyword description="@offersServiceInLanguage:
    English=English@area=Main Building@keyword=
    @hasTerrace=true@hasPrice=13@type=Traditional">
    </keyword>
  </title>
  <body>
    <text>Typical museum cafeteria restuarant seving
    day manus for families and assorted pintxos at a
    reasonable price. The restaurant offers terrace
    service on sunny wather. Located in the San
    Sebastian Technology Park Science Museum,
    it offers an wonderful atmosphere and an outdoor
    museum with minuature models of the
    Guipuzcoan Cultural Heritage.</text>
  </body>
</message>
<message>
  <objectparameters>
    <defaultlanguage iso="spa"></defaultlanguage>
  </objectparameters>
  <title>
    Arbelaitz1
    <keyword description="@offersServiceInLanguage:
    Spanish=Spanish@area=Main Building@keyword=
    @hasTerrace=false@hasPrice=30@type=Traditional">
    </keyword>
  </title>
  <body>
```

```
    <text>Arbelaitz1 is a very posh and elegant
    restaurant located at the centre of the San
    Sebastian Technology Park. It was awarded
    with two Michelin Stars and it is run by one
    of the most famous basque chefs. It offers
    a quiet atmosphere, cozy restaurant very
    appropriate for business lunch meeting.
    They serve a newly designed meny
    consisting of three courses, dessert,
    wine, bread and coffee. It is relatively
    expensive but good quality food
    are guaranteed.</text>
  </body>
</message>
<message>
  <objectparameters>
    <defaultlanguage iso="eng"></defaultlanguage>
  </objectparameters>
  <title>
    Arbelaitz2
    <keyword description=
    "@offersServiceInLanguage:
    English=English@area=Main Building@keyword=
    @hasTerrace=false@hasPrice=11@type=
    Traditional">
    </keyword>
  </title>
  <body>
    <text>Arbelaitz2 is the cheaper version of
    Arbelaitz1. Modern traditional food is served
    at this restaurant. It is also located at the
    centre of the San Sebastian Technology Park,
    very close to the main building. There is
    very good quality food and the atmosphere
    is also very comfortable. It is a very
    adequate restaurant to have lunch on
    your own or an informal lunch meeting.
    The menu consists of two course menu,
    drink, bread and dessert. They also serve
    assorted pintxos and sandwiches on
    demand. The service is relatively
```

```
      quick so it is also appropriate if somebody
      does not have too much time for lunch.
      </text>
    </body>
  </message>
  <message>
    <objectparameters>
      <defaultlanguage iso="spa">
      </defaultlanguage>
    </objectparameters>
    <title>
      Fast Food
      <keyword description=
      "@offersServiceInLanguage:
      Spanish=Spanish@area=
      Further End@keyword=
      @hasTerrace=true@hasPrice=7@type=
      Traditional">
      </keyword>
    </title>
    <body>
      <text>Este restaurante esta ubicado al final
      del Parque Tecnologico de San Sebastian.
      Es un restuarante de comida rapida en el
      que se sirven bocadillos, hamburguesas,
      refrescos, cervezas, etc. Esta pensado para
      gente que no dispone de demasiado tiempo
      para comer. El ambiente es agradable, tiene una
      terraza amplia en la que se ofrece servicio
      gratuito de conexion inalambrica a internet
      y el precio es muy asequible</text>
    </body>
  </message>
  <message>
    <objectparameters>
      <defaultlanguage iso="eng"></defaultlanguage>
    </objectparameters>
    <title>
      TechPark Vegetarian Restaurant
      <keyword description="@offersServiceInLanguage:
      Spanish=Spanish@offersServiceInLanguage:English=
```

```
    English@area=Further End@keyword=@hasTerrace=
    true@hasPrice=8@type=Vegetarian">
    </keyword>
  </title>
  <body>
    <text>Este es el único restaurante vegetariano
    en el Parque Tecnologico. Ofrece ensaladas,
    verduras, sandwiches vegetarianos, yogures,
    fruta, todo tipo de comida sana y verde.
    Refrescos, cerveza elaborada en sus
    instalaciones. Es un restaurante muy adecuado
    para personas que están de paso y que no
    disponen de demasiado tiempo para comer
    y que desean mantener la linea. Esta ubicado
    al final del parque tecnologico, en
    una zona de reciente construccion.
    Dispone de una amplia terraza, con vistas
    y con conexion gratuita a internet a
    traves de su conexion inalambrica.</text>
  </body>
</message>
<message>
  <objectparameters>
    <defaultlanguage iso="spa"></defaultlanguage>
  </objectparameters>
  <title>
    Mamistegi
    <keyword description=
    "@offersServiceInLanguage:
    Spanish= Spanish@area=Entrance@keyword=
    @hasTerrace=
    true@hasPrice=12@type=Modern"><
    /keyword>
  </title>
  <body>
    <text>Tipico restaurante de menu del dia.
    Menu consistente en un primer plato, un
    segundo plato, postre, pan, bebida,
    refrescos, postre y cafe.
    Tiene terraza y el menu cuesta en
    torno a los 12 euros.</text>
```

```
    </body>
  </message>
  <message>
    <objectparameters>
      <defaultlanguage iso="eng"></defaultlanguage>
    </objectparameters>
    <title>
      Valles
      <keyword description="@offersServiceInLanguage:
      Spanish=Spanish@offersServiceInLanguage:
      English=
      English@area=Entrance@keyword=
      @hasTerrace=true@hasPrice=15@type=Traditional">
      </keyword>
    </title>
    <body>
      <text>Traditional Restaurant in Donostia.
      It has a more than 75 year long tradition
      in serving traditional pintxos, portions,
      sandwiches.</text>
    </body>
  </message>
  <message>
    <objectparameters>
      <defaultlanguage iso="eng"></defaultlanguage>
    </objectparameters>
    <title>
      Basque Traditional Tavern
      <keyword description="@offersServiceInLanguage:
      English=English@area=Entrance@keyword=
      @hasTerrace=false@hasPrice=20@type=Traditional">
      </keyword>
    </title>
    <body>
      <text>Basque traditional restaurant. Day menus
      and assorted pintxos are served, with typical
      basque drinks and food cooked at a
      traditional way</text>
    </body>
  </message>
</menu>
```

```
</journaline>
```

Appendix IV

Forth Appendix: The User Evaluation Questionnaire

This appendix shows the questionnaire that has been used in the user evaluation. As has been explained in Chapter 6, the TAM model has been used for this purpose[Dav89].

The user evaluation has been divided into four different sections, namely Attitude (AT), Perceived Usefulness (PU), Perceived Ease of Use (PEOU) and Behavioural Intention (BI). A set of questions have been designed for each of these constructs, based on the literature. A range from 1 to 4 has been offered to participants to answer, 1 being "completely disagree" and 4 being "completely agree".

In order to have more sensible results on the questionnaires, all of the participants had to answer three questions related to the moderating factors used, in order to increase the explicability of the reuslts of the questionnaire. The moderating factors are:

- Gender.

	Male	Female
Gender		

Table IV.1: Moderating factor: Gender

- Age.

	20 - 25	26 - 30	31 - 35	36 - 40	41 - 45	46 - 50	50 +
Age							

Table IV.2: Moderating factor: Age

- Educational Background.

	University Degree	ABD	PhD
Educational Background			

Table IV.3: Moderating factor: Educational Background

IV.a Attitude

The AT construct refers to *"readiness of a person to use a particular technology"*.

	1	2	3	4
I like using the CONCERT framework				
Using the CONCERT framework is fun				
CONCERT provides an attractive user interface				
I frequently use technology				
I had enough prior knowledge as to use CONCERT				
The indications I have received have been adequate to appropriately use CONCERT				

Table IV.4: Set of questions to measure AT

IV.b Perceived Usefulness

The PU is the *"the degree to which a person believes that using a particular technology will enhance his performance*[Dav89].

	1	2	3	4
Using the CONCERT framework based applications supported me on the move				
Using the CONCERT framework based applications improved my tourism experience				
Using the CONCERT framework based applications allowed me to move in the place more efficiently				
Using the CONCERT framework based applications made it easier for me to find what I needed				

Table IV.5: Set of questions to measure PU

IV.c Behavioural Intention

	1	2	3	4
I would recommend friends to use the CONCERT framework based application in their trips				
I would recommend the implementation of the CONCERT framework in future applications				

Table IV.6: Set of questions to measure BI

IV.d Perceived Ease of Use

The PEOU is the *"degree to which a person believes that using a particular system would be free from effort*[Dav89].

	1	2	3	4
It is easy to provide personal information of context to the CONCERT framework based application				
It is easy to get the CONCERT framework based application to provide context-based information				
Interaction with the CONCERT framework based application was clear and understandable				
Getting information from the CONCERT framework based application was easy				
Learning how to use CONCERT framework based application was easy				
Becoming a CONCERT framework based application skilful user is easy				

Table IV.7: Set of questions to measure PEOU

Appendix V

Fifth Appendix: System Operator

TRIZ[1] is a set of tools, methods and a new way of thinking to generate innovative ideas and solutions for problem solving. TRIZ provides tools and methods for use in problem formulation, system analysis, failure analysis and patterns of system evolution. It aims at creating an algorithmic approach to the invention of new systems. The figure below (fig. V.1) shows that at the very highest level, TRIZ may be seen as the systematic study of excellence. This study was initially focused on patents (a very good source of excellence for the most part), and then evolved to look at excellence in the sciences, and latterly, the arts, business, social sciences and politics.

As shown in the picture, at the bottom of the pyramid, there are a wide-range and comprehensive series of tolls and techniques that are used in most of the cases for problem solving analysis. The tools contain a great - some might say overwhelming- level of richness and to all intents and purposes, it may be said that there is a tool for practically any problem or opportunity situation that may be encountered. One of these tools is the System Operator or 9-Windows, which has been widely used through out this dissertation in order to more thoroughly think about the notion of context within the Human Mobility framework.

[1]Romanized acronym in English "The theory of inventor's problem solving"

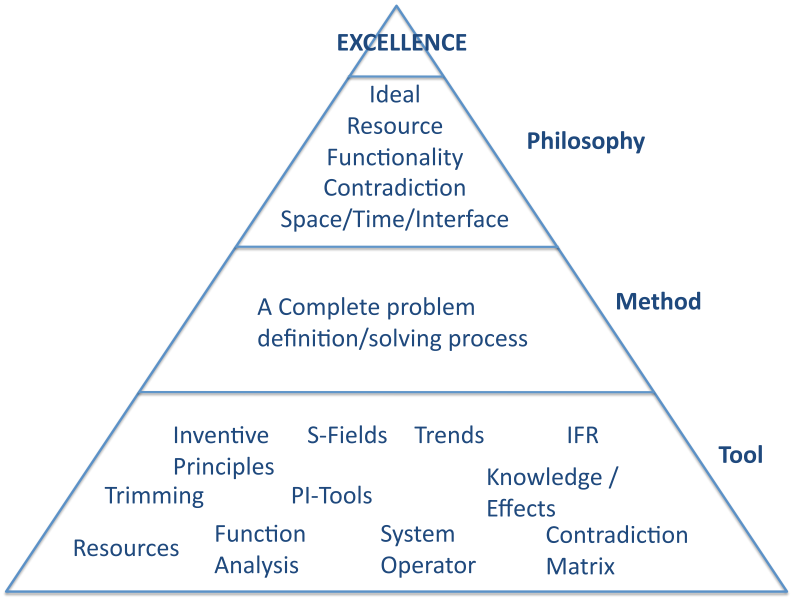

Figure V.1: Hierarchical view of TRIZ. (Source: Mann).

V.a System Operator

The System Operator tool (or 9-Windows) is a simple means of helping people think in terms of TIME and SPACE. The basic principle of operation divides the world into nine segments as shown in figure V.2.

The figure represents the following:

- *Present*: The current moment.
  - *System*: The central box of the nine is the one humans' brains naturally migrate to whenever they are given a problem situation. In other words, asked to think about "designing a better pen", for example, humans' brains are likely to immediately conjure up the image of a pen, i.e. the system, being used to write ("the present").
  - *Super-system*: What the system operator is trying to get people to do is also to think about the pen in a bigger ("super-system") context, for example, a person holding the pen, the ink molecules, and so on.
  - *Sub-system*: The smaller ("sub-system") context, i.e. the components of the pen, the paper, the desk, and so on.

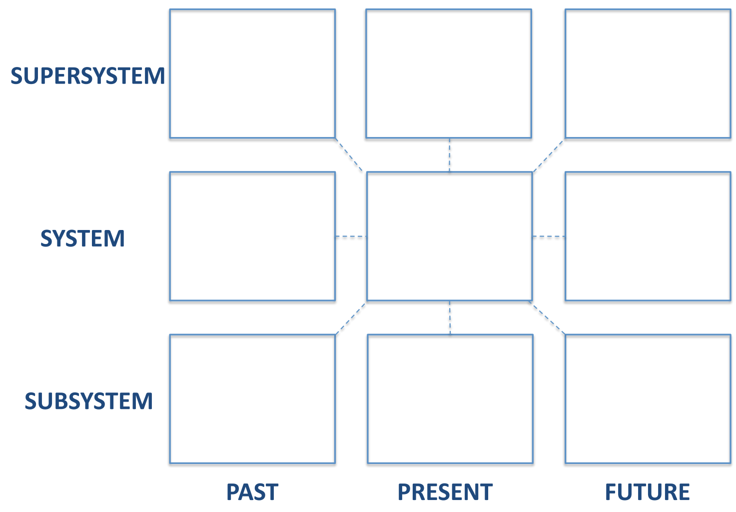

Figure V.2: The System Operator. (Source: Mann).

- *Past*: what happened to the pen prior to the moment considered, e.g. manufacturing, shipping, and so on.
- *Future*: what happens to the pen immediately after one has finished writing, right through its disposal after it has run out.

Figure V.3 illustrates more in detail the main time and space features that may be considered when thinking more completely about the design of a pen. The point of this tool is to help people overcome the psychological inertia of present and system level only. In other words, to concentrate on the problem itself, on analyzing all of its possible aspects, without automatically running to the brain comfort-zone, which is that of problem solving. It is needed a tool to make humans think more deeply in the problem, and the 9 Windows is one very powerful one.

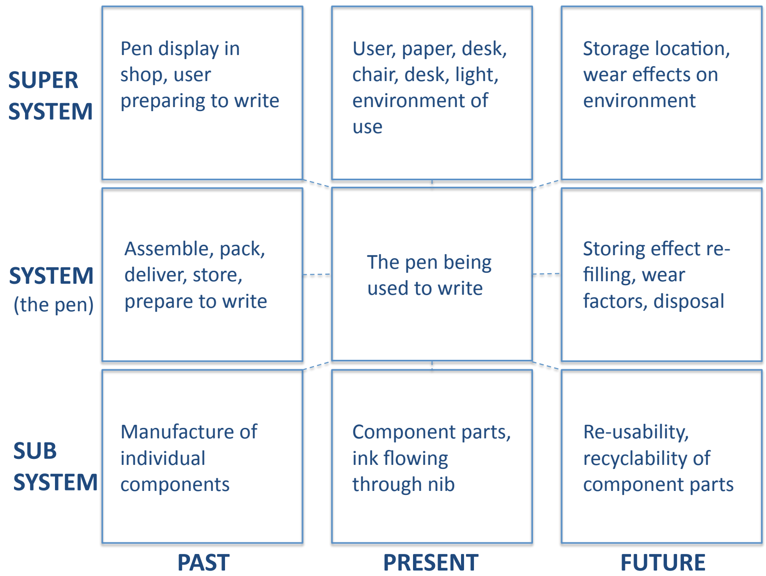

Figure V.3: Ths System Operator for “pen”. (Source: Mann).

References

[AAH+97] Gregory D. Abowd, Christopher G. Atkenson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Beltzer Journals*, September 1997.

[ABB+09] Akinori Abe, Victoria Beltran, Jean-Pierre Briot, Pascal Bouvry, Jord Casademont, Marisa Catalan, Yasmine Charif, Waltenegus Dargie, Franco Davoli, Jianguo Ding, and Amal El-Fallah-Seghrouchni. *Context-Aware Computing and Self-Managing Systems*. CRC Press, 2009.

[Abd07] Ibrahim Abdalla. Evaluating effectiveness of eblackboard system using tam framework: A structural analysis approach. *Assocaition for the Advancement of Computing in Education Journal (AACEI Journal)*, 15(3):279 – 287, 2007.

[ABR05] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. Loosely coupling ontological reasoning with an efficient middleware for context-awareness. In *Mobile and Ubiquitous Systems: Networking and Services*, pages 175 – 182, July 2005.

[ADG+05] Grigoris Antoniou, Carlos Viegas Damasio, Benjamin Grosof, Ian Horrocks, Michael Kifer, Jan Maluszynski, and Peter F. Patel-Schneider. Combining rules and ontologies. a survey. Technical report, March 2005.

[AQFHS07] Muhammad Abdul-Qadir, Muhammad Fahad, and Syed-Adnan Hussain-Shah. Incompleteness errors in ontologies. In *Proceedings of the International Conference on Granular Computing*, 2007.

[Ari C] Aristotle. *Metaphysics IV*. 3 b. C.

[Ay07] Feruzan Ay. Context modelling and reasoning using ontologies. 2007.

[Bal07] Matthias Baldauf. A survey on context-aware systems. *International Journal Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[Bat07] Fabio Battussi. *A User-Adaptive and Context-Aware Architecture for Mobile and Desktop Training Applications*. PhD thesis, September 2007.

[BBH+09] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 2009. doi:10.1016/j.pmcj.2009.06.002.

[BBWA97] Pim Borst, Jan Benjamin, Bob Welinga, and Hans Akkermans. An application of ontology construction. *International Journal of Human and Computer Studies*, 46(2 – 3):365 – 406, 1997.

[BCCGP02] V. Richard Benjamins, Jesús Contreras, Oscar Corcho, and Asunción Gómez-Pérez. Six challenges for the semantic web. In *In KR2002 Semantic Web Workshop*, 2002.

[BCGB07] Ana M. Bernardos, José R. Casar Corredera, Javier I. Portillo García, and Ana B. Bermejo. Informe de vigilancia tecnológica: Servicios y aplicaciones en movilidad para el sector turístico. Watchtower, Universidad Politécnica de Madrid, 2007.

[BCK05] Jrg Baus, Keith Cheverst, and Christian Kray. A survey in map-based mobile guides. In Liqui Meng and Alexander Zipf, editors, *Map-based mobile guides: theories, methods and implementations*, pages 197 – 216, 2005.

[Bee09] Thomas Beer. *Design and Implementation of a Context-Aware Information Push System (CAIPS)*. PhD thesis, University of Innsbruck, March 2009.

[BFH+07] Thomas Beer, Matthias Fuchs, Wolfram Höpken, Jörg Rasinger, and Hannes Werthner. Caips: A context-aware information push service in tourism. In Marianna Sigala, Luisa Mich, and Jamie Murphy, editors, *Informatin and Communication Technologies in Tourism*, pages 129 – 140, 2007.

[BFLGPGP98] Mercedes Blázquez, Mariano Fernández-López, Juan Manuel García-Pinar, and Asunción Gómez-Pérez. Building ontologies at the knowledge level using the ontology design environment. In B.R. Gaines and M.A. Musen, editors, *11th Internation Conference on Knowledge Acquisition, Modelling and Management*, 1998.

[BFP+09] Robert Barta, Christina Feilmayr, Birgit Pröll, Christoph Grün, and Hannes Werthner. Covering the semantic space of tourism an approach based on modularized ontologies. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, number ISBN:978-1-60558-528-4, 2009.

[BLC96] Amaia Bernaras, Inaki Laresgoiti, and Jose Corera. Building and reusing ontologies for electrical network and applications. In W. Washler, editor, *In Proceedings of the 12th European Conference on Artificial Intelligence*, 1996.

[BM08] D. Brickley and L. Miller. Foaf vocabulary specification 0.9. Technical report, 2008.

[BMU02] Sigrid Steinholt Bygdås, Pål S. Malm, and Tore Urnes. A simple architecture for delivering context information to mobile users. In *Workshop at Handheld and Ubiquitous Computing*, 2002.

[BP08] Dimitrios Buhalis and Lucca Pistidda. The impact of wimax on tourist destinations. In *Inproceedings of the ENTER conference*, pages 383 – 394, 2008.

[Bro98] P. J. Brown. Triggering information by context. *Personal Technologies*, 2(1):1 – 9, 1998.

[Buh03] Dimitrios Buhalis. *Tourism: information technology for strategic tourism management.* Pearson (Financial Times/Prentice Hall), 2003.

[CDM+00] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of the CHI*, 2000.

[CF03] Harry Chen and Tim Finin. An ontology for context aware pervasive computing environments. In *Inproceedings of the IJCAI Workshop on Ontologies and Distributed Systems*, 2003.

[CFJ04a] Harry Chen, Tim Finin, and Anupam Joshi. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review, Cambridge University Press, Vol. 18, pp.197207*, 18:197 – 207, 2004.

[CFJ04b] Harry Chen, Tim Finin, and Anupam Joshi. Semantic web in the context broker architecture. In *Proceedings of the Second Annual IEEE International Conference on Pervasive Computer and Communications*. IEEE Computer Society, 2004.

[CFJ04c] Harry Chen, Tim Finin, and Anupan Joshi. Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing*, pages 2–12, 2004.

[CFJ05] Harry Chen, Tim Finin, and Anupam Joshi. *Ontologies for Agents: Theory and Experiences*, chapter The SOUPA ontology for Pervasive Computing. Whitestein Series in Software Agent Technologies. Springer, July 2005.

[CFLGP01] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Ontoweb: Technical roadmap. Deliverable 1.1, Universidad Politécnica de Madrid, November 2001.

[CFLGP07] Oscar Corcho, Mariano Fernández-López, and Asunci'on Gómez-Pérez. *Ontological Engineering: What are Ontologies and How can We build them?*, chapter Chapter III:, pages 44–70. Idea Group Inc., 2007.

[CGK05] Eleni Christopoulou, Christos Goumopoulos, and Achilles Kameas. An ontology-based context management and reasoning process for ubicomp applications. In *In Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, volume 121, pages 265 – 270, 2005. ISBN:1-59593-304-2.

[Che03] Harry Chen. A proposal for defining pervasive computing standard ontology (percom-so). *Knowledge Engineering Review*, 18(3):197 – 207, 2003.

[Che04] Harry L. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, Department of Computer Science and Electrical Engineerin, University of Baltimore, 2004.

[Che06] Thomas Chesney. An acceptance model for useful and fun information systems. *An Interdisciplinary Journal on Humans in ICT Environments*, 2(2):225 – 235, October 2006.

[CJB99] B. Chandrasekaran, John R. Josephson, and Richard Benjamins. What are ontologies and why do we need them? *IEEE Intelligent Systems*, pages 20–26, 1999.

[CK01] Guanling Chen and Kevin Kotz. A survey on context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, November 2001.

[CK03] Catharina Candolin and Hannu H. Kari. An architecture for context-aware management. In *Proceedings of IEEE MILCOM*, 2003.

[CMN03] Wynne W. Chin, Barbara L. Marcolin, and Peter R. Newsted. A partial least squares latent variable modeling approach for measuring interaction effects: results from a monte carlo simulation study and an electronic mail emotion/adoption study. *Information Systems Research*, 14(s):21 – 41, 2003.

[Coh95] Paul R. Cohen. *Empirical Methods for Artificial Intelligence*. The MIT Press, 1995.

[CPC+04] Harry Chen, Filip Perich, Dipanjan Chakraborty, Tim Finin, and Anupam Joshi. Intelligent agents meet semantic web in a smart meeting room. *Thir international joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 854 – 861, 2004.

[CRL+09] Alejandro Cadenas, Carlos Ruiz, Iker Larizgoitia, Raúl García-Castro, Carlos Lamsfus, Iñaki Vázquez, Marta González, David Martín, and María Poveda. Context management in mobile environments: a semantic approach. In *In proceedings of the Workshop on Context, Information and Ontologies of the European Semantic Web Conference*, ISBN:978-1-60558-528-4, June 2009.

[CVGL09] Alejandro Cadenas, Inaki Vázquez-Gómez, and Iker Larizgoitia. Informe de requisitos de gestión de contexto en un entorno móvil. Technical Report E1.2.1, TID and UD, March 2009.

[DA00a] Anind K. Dey and Gregory Abowd. The context toolkit: Aiding the development of context-aware applications. In *Workshop on Software Engineering for Wearable and Pervasive Computing*, 2000.

[DA00b] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Proceedings of the Workshop on the What, Who, Where, When and How of Context Awareness*. ACM Press, 2000.

[DAS99] Anind K. Dey, Gregory D. Abowd, and D. Salber. Context-based infrastructure for smart environments. In *1st International Workshop on Managing Interactions in Smart Environments (MANSE'99)*, pages 114 – 128, 1999.

[Dav89] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13:319 – 340, 1989.

[DBW92] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology*, 22(14):1111 – 1132, 1992.

[DC04] Patricia Dockhorn-Costa. Towards a service platform for mobile context-aware applications. In *Ubiquitous Computing - IWUC*, pages 48–61, 2004.

[DC07] Patricia Dockhorn-Costa. Towards conceptual foundations for context-aware applications. In *Proceeding of the 2007 conference on Advances in Ambient Intelligence*, pages 36 – 55, 2007.

[DCG+89] Peter J. Denning, Douglas E. Comer, David Gries, Michael D. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. Computing as a discipline. *Communications of the ACM*, 32(1):9 – 29, 1989.

[DCGP+06] Patricia Dockhorn-Costa, Giancarlo Guizzardi, Joao Paulo, A. Almeida, Luís Ferreira Pires, and Marten Van Sinderen. Sinderen. situations in conceptual modeling of context. In *In EDOC 2006 workshop proceedings*, pages 6 – 16, 2006.

[Den81] Peter J. Denning. Performance evaluation: Experimental computer science at its best. *Communications of the ACM, This is the text of the ACM Presidentś Letter for the Communications of ACM*, 1981.

[Den03] Peter J. Denning. Great principles of computing. *Communications of the ACM*, 46(10):15 – 20, 2003.

[Den05] Peter J. Denning. Is computer science science? *Communications of the ACM*, 48(4):27 – 31, 2005.

[Dey00] Anind K. Dey. *Providing Architectural Support for Building Context-Aware Applications.* PhD thesis, Georgia Institute of Technology, November 2000.

[Dey01] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4 – 7, February 2001.

[DP98] T. H. Davenport and L. Prusak. *Working knowledge: how organizations manage what they know.* Harvard Business School Press, 1998.

[DSAF99] Anind K. Dey, Daniel Salber, Gregory D. Abowd, and Masayasu Futakawa. An architecture to support context-aware applications. *12th Annual ACM Symposium on User Interface Software and Technology*, 1999.

[dSSRC+07] Luiz Olavo Bonino da Silva Santos, Fano Ramparany, Patricia Dockhorn Costa, Peter Vink, Richard Etter, and Tom Broens. A service architecture for context awareness and reaction provisioning. In *IEEE Congress on Services*, 2007.

[EOSB03] M. Eisenhauer, R. Oppermann, and B. Schmidt-Belz. Mobile information systems for all. In *In proceedings of the Tenth International Conference on Hunan-Computer interaction*, pages 354 – 358, 2003.

[ETS05a] ETSI. Digital radio mondiale (drm); distribution and communications protocol (dcp). Technical Specification ETSI TS 102 821 V1.2.1 (2005-10), ETSI - European Telecommunications Standards Institute, 2005.

[ETS05b] ETSI. Digital radio mondiale (drm); multiplex distribution interface (mdi). Technical Specification ETSI TS 102 820 V1.2.1 (2005-10), ETSI - European Telecommunications Standards Institute, 2005.

[ETS05c] ETSI. Digital radio mondiale (drm); system specification. Technical Specification ETSI ES 201 980 V2.2.1 (2005-10), ETSI - European Telecommunications Standards Institute, 2005.

[ETS08] ETSI. Digital audio broadcasting (dab). journaline. user specification. Technical Specification ETSI TS 102 979 V1.1.1, ETSI - European Telecommunication Standards Institute, June 2008.

[Euz95] Jerome Euzenat. Building consensual knowledge bases: Context and architecture. In N. Mars, editor, *Second International Conference on Sharing of very Large-Scale Knowledge Bases (KBKS'95)*, pages 143 – 155, 1995.

[Euz96] Jerome Euzenat. Corporate memory through cooperative creation
of knowledge bases and hyper-documents. In B.R. Gaines and M. A. Musen, editors, *10th Knowledge Acquisition for Knowledge-based Systems Workshop (KAW'96)*, 1996.

[Fei06] Dror G. Feitelson. Experimental computer science: The need for a cultural change. *Experimental Computer Science: The Need for a Cultural Change*, 2006.

[FHKL09] David Faveur, Manfred Hackl, Marc Wilhelm Küster, and Carlos Lamsfus. Harmonization of data interchange in tourism. Technical report, CEN/ISSS, 2009.

[FI08] Fraunhofer-IIS. Drm content server - user manual. User Manual 4.0.62.6681.2008-11-13, Fraunhofer Geselschaft, November 2008.

[GF95] Michael Grüninger and Michael Fox. Methodology for design and evaluation of ontologies. In *In IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[GG95] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases. In In N. Mars, editor, *Towards very large Knowledge bases: Knowledge Building and Knowledge Sharing*, pages 25 – 32, 1995.

[GHF06] Ulrike Gretzel, Y. H. Hwang, and Daniel R. Fesenmaier. *Destination Recommendation Systems*. CABI, 2006.

[GPFLC03] Asunción Gómez-Perez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering*. Springer Verlag, 2003. ISBN 1-85233-551-3.

[GPFLP97] Asuncion Gómez-Pérez, Mariano Fernández-López, and J. Pazos. *Ingeniería del Conocimiento: Diseño y Construcción de Sistemas Expertos*. Ceura, 1997.

[GPSF09] Asunción Gómez-Perez and Mari-Carmen Suárez-Figueroa. Scenarios for building ontology networks within the neon methodology. In *In proceedings of the Knowledge Capture Conference*, pages 183 – 184, 2009.

[GPW+08] Christoh Grün, Birgit Pröll, Hannes Werthner, Werner Retschitzegger, and Wieland Schwinger. Assisting tourists on the move: An evaluation of mobile tourist guides. In IEEE Computer Society, editor, *7th International Conference on Mobile Business*, pages 171 – 180, 2008.

[GPZ94] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A middleware for building context-aware mobile services. In *In Proceedings of IEEE Vehicular Technology Conference*, 1994.

[GPZ04] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A bayesian approach for dealing with uncertain contexts. In *Proceedings of the Second International Conference on Pervasive Computing*, volume 176. Austrian Computer Society, 2004.

[GPZ05] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28:1–18, 2005.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199 – 220, 1993.

[Gru94] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *In Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, in press. Substantial revision of paper presented at the International Workshop on Formal Ontology, 1994.

[Gua95] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human and Computer Studies*, 43:625 – 640, 1995.

[GW04] Ulrike Gretzel and Karl Wöber. Intelligent search support: Building search term associations for tourism-specific search engines. In Andy Frew, editor, *In Proceedings of the Information and Communication Technologies in Tourism*, pages 239 – 248, 2004.

[GWPZ04] Tao Gu, Hang Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004.

[GYF00] Ulrike Gretzel, Yu-Lan Yuan, and Daniel R. Fesenmaier. Preparing for the new economy: Advertising strategies and change in destination marketing organizations. *Journal of Travel Research*, 39(2):146 – 156, 2000.

[HÖ8] Wolfram Höpken. Context-based adaptation of ubiquitous web applications in tourism. In Ulrike Gretzel Peter O'Connor, Wolfram Hoepken, editor, *Information and Communication Technologies in Tourism 2008*, 2008.

[Har] Harper. People vs. information: the evolution of mobile technology. Technical report, University of Surrey.

[HB06] Annika Hinze and George Buchanan. Context-awareness in mobile tourist information systems: Challenges for user interaction. In *In Proceedings of the ACSC06, Australian Computer Society*, pages 257–266. Society, Inc, 2006.

[HBSS02] Albert Held, Sven Buchholz, Alexander Schill, and Er Schill. Modeling of context information for pervasive computing applications. In *Modeling of Context Information for Pervasive Computing Applications*, 2002.

[HFZ+06] Wolfram Höpken, Matthias Fuchs, Markus Zanker, Thomas Beer, Alexander Eybl, Stefan Flores, Segiu Gordea, Markus Jessenitschning, Thomas Kerner, Dirk Linke, Jörg Rasinger, and Michael Schnabl. etplanner: An it framework for comprehensive and integrative travel guidance. In Martin Hitz, Marianna Sigala, and Jamie Murphy, editors, *Information and Communication Technologies in Tourism*, pages 125 –134, 2006.

[HIR02] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive compututing systems. In Lecture Notes in Computer Science, editor, *1st International Conference on Pervasive Computing (Pervasive)*, volume 2414, pages 167 – 180. Springer Verlag, 2002.

[HL09] Ramon Hervás-Lucas. *Modelado de Contexto Para la visualización de información en ambientes inteligentes*. PhD thesis, Universidad de Castilla La Mancha, February 2009.

[HMP04] Alan R. Hevner, Salvatore T. March, and Jinsoo Park. Design science in information systems research. *MIS Quarterly*, 28(1):75 – 106, March 2004.

[HNBR97] Richard Hull, Philip Neaves, and James Bedford-Roberts. Towards situated computing. In *In Proceedings of the IEEE ISWC - First International Symposium on Wearable Computers*, 1997.

[Hob02] Jerry Hobbs. Daml-time ontology. Technical report, 2002.

[Hoe] Wolfram Hoepken. etplanner presentation.

[HP06] Jerry R. Hobbs and Feng Pan. Time ontology in owl. Technical report, W3C, September 2006.

[HRW+06] P. Haase, S. Rudolph, Y. Wang, S. Brockmans, R. Palma, J. Euzenat, and M. d'Aquin. Networked ontology model. Technical Report - Deliverable D1.1.1, Uni Karlsruhe, November 2006.

[HSB+05] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilanmowitz-Mllendorf. Gumo - the general user modelling ontology. In *In User Modelling*, pages 428 – 432, 2005.

[HSLF08] Wolfram Hoepken, Markus Scheuringer, Dirk Linke, and Matthias Fuchs. Context-based adaptation of ubiquitous web applications in tourism. In Peter O'Connor, Ulrike Gretzel, and Wolfram H"opken, editors, *Information and Communication Technologies in Tourism*, pages 535 – 544, 2008.

[ICB09] Alessandro Inversini, Lorenzo Cantoni, and Dimitrios Buhalis. Destinations' information competition and web reputation. *Journal of Information Technology and Tourism*, 11:221 – 234, 2009.

[IEE90] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Computer Society, 1990.

[IEE96] IEEE. *IEEE Standard for Developing Software Life Cycle Processes*. IEEE, 1996. IEEE Standard 610.1074-1995.

[IEE97a] IEEE. *IEEE Standard for Developing a Software Project Life Cycle Process. IEEE Std 1074-2006*. IEEE, 1997.

[IEE97b] IEEE. *IEEE Standard for Developing Software Life Cycle Processes. IEEE Std 1074-1997*. IEEE, 1997.

[Inc07] Qualcomm. Incorporated. The evolution of wireless applications and services. Technical report, Qualcomm Incorporated, 2007.

[JMSK09] Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, 158:1 – 18, 2009.

[Kah05] Kevin Kahn. The evolution of wimax. Technical report, Intel Corporation, 2005.

[KB03] C. Kray and Jrg Baus. A survey of mobile guides. In *In workshop HCI at Mobile HCI*, 2003.

[KFJ03] L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 63 – 74, 2003.

[Kli05] Alejandro Klimovski. *Las desventuras del conocimiento cientifico*. Ed. A.Z., 2005. ISBN 950-534-275-6.

[KNC07] Judy Kay, William T. Niu, and David J. Carmichael. Oncor: Ontology- and evidence-based context reasoner. In *In proceedings of the IUI*, pages 290 – 293. ACM Press, January 2007.

[KPM05] Anders Kofod-Petersen and Marius Mikalsen. An architecture supporting implementation of context-aware services. In *Workshop on Context Awareness for Proactive Systems (CAPS 2005)*, pages 31–42. HIIT Publications, 2005.

[KS07] Reto Krummenacher and Thomas Strang. Ontology-based context modeling. In *In Workshop on Context-Aware Proactive Systems*, 2007.

[LAS09] Carlos Lamsfus and Aurkene Alzua-Sorzabal. Computación contextual basada en tecnologias semanticas en el ambito de la movilidad humana. *CIC Network*, (6):50 – 54, 2009.

[LASM+09a] Carlos Lamsfus, Aurkene Alzua-Sorzabal, David Martín, Zigor Salvador, and Alex Usandizaga. Contextual computing based services in tourism. In *In Proceedings of the 4th Mediterranean Conference on Information Systems*, pages 677 – 690, September 2009.

[LASM+09b] Carlos Lamsfus, Aurkene Alzua-Sorzabal, David Martín, Zigor
Salvador, and Alex Usandizaga. Human-centric ontology-based context modelling in tourism. In *In Proceedings of the*

1st International Conference on Knowledge Engineering and Ontology Development, 2009.

[LCG08] Dong Liu, Junliang Chen, and Jie Guo. An ontology-based lbs system. In *In Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing 2008*, pages 1 – 4, 2008.

[Len95] Douglas B. Lenat. Cyc: A large scale investment in knowledge infrastructure. *Communications of ACM*, 38(11):32 – 38, November 1995.

[LG90] D. B. Lenat and R. V. Guha. *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley, 1990.

[Man02] Darrell Mann. *Hands on Systematic Innovation.* IFR, 2002.

[MCM03] R. E. McGrath, R. H. Campbell, and M. D. Mickunas. Ontologies in a pervasive computing environment. In *Workshop on Ontologies in Distributed Systems, IJCAI*, pages 10 – 14. ACM Press, 2003.

[MH69] John Mccarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463 – 502. Edinburgh University Press, 1969.

[MRCM03] R.E. McGrath, A. Ranganathan, R. H. Campbell, and M. D. Mickunas. Use of ontologies in pervasive computing environments. Technical Report UIUCDCS-R-2003- 2332 ULU-ENG-2003-1719, 2003.

[Mus92] Mark A. Musen. Dimensions of knowledge sharing and reuse. *Computers and Biomedical Research*, 25:435 – 467, 1992.

[New80] Allen Newell. The knowledge level (presidential address). *AI Magazine*, 2(2):1 – 20, 1980.

[NFF+91] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):36 – 56, 1991.

[Nic08] Daniela Nicklas. Context modelling and reasoning - key concepts for pervasive computing, March 2008.

[NM01] Natasha F. Noy and Deborah MacGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL 01-05, Stanford University, 2001.

[OM10] Peter O'Connor and Jamie Murphy. Hotel domain name adoption and implementation. In *In Proceedings of Information and Communication Technologies in Tourism 2010*, 2010.

[Pas98] Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 92 – 99, 1998.

[Pas01] Jason Pascoe. *Context-Aware Software*. PhD thesis, Computing Laboratory, University of Kent at Canterbury, 2001.

[Pas97] Jason Pascoe. The stick-e note architecture: Extending the interface beyond the user. pages 261–264. ACM Press, 97.

[PBHS03] Ariel Pashtan, Remy Blattler, Andi Heusser, and Peter Scheuermann. Catis: A context-aware tourism information system. In *In proceedings of the 4th International Workshop on Mobile Computing*, June 2003.

[PdBW+06] Davy Preuveneers, J.V. den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K.D. Bosschere. Towards an extensible context ontology for ambient intelligence. *In Proceedings of 2nd European Symposium Ambient Intelligence, LNS 3295*, pages 148 – 159, 2006.

[Poo03] Auliana Poon. *Tourism, Technology and Competitive Strategies*. CAB Imtermational, 2003.

[PS07] Jhoanna Rhodette Pedrassa and Aruna Seneviratne. A proposed architecture for context-aware mobility management. In *2007 International Symposium on Communications and Information Technologies (ISCIT 2007)*, pages 729–734, 2007.

[PTS04] H.S. Pinto, C. Tempich, and S. Staab. Diligent: To- wards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *16th European Conference on Artificial Intel- ligence (ECAI 2004)*, pages 393 – 397, 2004.

[PVR+06] Davy Preuveneers, Yves Vandewoude, Peter Rigole, Dhouha Ayed, and Yolande Berbers. Context-aware adaptation for component-based pervasive computing systems. 2006.

[Rib06] Daniele Riboni. *Techniques, Algorithms, and Architectures for Adaptation in Mobile Computing*. PhD thesis, Universtià degli Studi di Milano, 2006.

[RJSFG06] Danny Raz, Arto Juhola, Joan Serrat-Fernandez, and Alex Galix. *Fast and Efficienc Context-Aware services*. Wiley Series in Communications Networking & Distributed Systems, 2006.

[Ros98] Paul S. Rosenbloom. A new framework for computer science and engineering. *IEEE Computer*, pages 32 – 40, 1998.

[RPB10] Daniele Riboni, Linda Pareschi, and Roberto Bettini. Towards the adaptive integration of multiple context reasoners in pervasive computing environments. In *In Proceedings of the 2010 CoMoRea Workshop*, pages 25 – 29, 2010.

[RPM97] N. Ryan, J. Pascoe, and D. Morse. Enhanced reality fieldwork: the context-aware archaeological assistent. In *Proceeding of the 25th Anniversary Computer Applications in Archaeology*, 1997. http://www.caaconference.org/.

[RSC07] Kasim Rehman, Frank Stajano, and George Coulouris. An architecture for interactive context-aware applications. *IEEE Pervasive Computing. Mobile and Ubiquitous Systems*, 6(1):73–80, January-March 2007.

[SAW94] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.

[SBF98] R. Studer, R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *IEEE Transactions on Data and Knowledge Engineering*, 25:161 – 197, 1998.

[Sch99] Albrecht Schmidt. There is more context than location. In *Computer & Graphics*, volume 23, pages 893–901, 1999.

[Sch02] Albrecht Schmidt. *Ubiquitous Computing - Computing in Context*. PhD thesis, University of Lancaster, November 2002.

[Sch06] Andreas Schmidt. Ontology-based user context management: The challenges of imperfection and time-dependence. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. Part I., ser. Lecture*, pages 995 – 1011. Springer Verlag, 2006.

[SDA99] Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 434 – 441, 1999.

[SF10] Roland Schegg and Michael Fux. A comparative analysis of content in traditional survey versus hotel review websites. In *In Proceedings of the Information and Communication Technologies in Tourism 2010*, 2010.

[SFdCB+07] Mari Carmen Suárez-Figueroa, Guadalupe Aguado de Cea, Carlos Buil, Caterina Caracciolo, Martin Dzbor, Asunción Gómez-Pérez, German Herrrero (ATOS), Holger Lewen, Elena Montiel-Ponsoda, and Valentina Presutti. Neon development process and ontology life cycle. Tech Report NEON/2007/D5.3.1/v1.0, UPM, August 2007.

[SFdCB+08] Mari Carmen Suárez-Figueroa, Guadalupe Aguado de Cea, Carlos Buil, Klaas Dellschaft, Mariano Fernández-López, Andrés García, Asunción Gómez-Pérez, German Herrero, Elena Montiel-Ponsoda, Marta Sabou, Boris Villazon-Terrazas, and Zheng Yufei. Neon: A methodology for building contextualized ontology networks. Technical Report D5.4.1, UPM, February 2008.

[SFFLGP+08] Mari Carmen Suárez-Figueroa, Mariano Fernández-López, Asunción Gómez-Pérez, Klaas Dellschaft, Holger Lewen, and Martin Dzbor. Neon: Revision and extension of the neon development process and ontology life cycle. NEON EU-IST-2005-027595 D5.3.2, UPM, November 2008.

[SFGP08] Mari-Carmen Suárez-Figueroa and Asunción Gómez-Pérez. Towards a glossary of activities in the ontology engineering field. In *In Proceedings of 6th International Conference on Language Resources and Evaluation*. UPM, May 2008.

[SGK05] Norman M. Sadeh, Fabian L. Gandon, and Oh Byung Kwon. Ambient intelligence: The mycampus experience. Technical Report CMU-ISRI-05-123, School of Computer Science. Carnegie Mellon University, School of Computer Science Carnegie Mellon Univerity Pittsburgh, PA 15213, July 2005.

[SGPR09] W. Schwinger, C. Grün, B. Pröll, and W. Restschitzegger. Context-awareness in mobile tourist guides. In *Handbook of Research on Mobile Multimedia*. Second edition information science reference edition, 2009.

[She97] Pauline Sheldon. Tourism information technology. *Journal of Travel Research*, 39(2), 1997.

[Sig06] Marianna Sigala. Mass customisation implementation models and customer value in mobile phones services. preliminary findings from greece. *Managing Service Quality*, 16(4):395 – 420, 2006.

[SIT09] SITA. 10 technology advantages that will change air travel. Technical report, SITA - - Specialists in Air Transportation Communiations and IT solutions, 2009.

[SL01] Albrecht Schmidt and Kristof Van Laerhoven. How to build smart appliances. *IEEE Personal Communications*, 8:66 – 71, 2001.

[SLP04] Thomas Strang and Claudia Linnhoff-Popien. A context modelling survey. In *In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[SLPF03] Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. Cool: A context ontology language to enable contextual interoperability. In *LNCS 2893: Proceedings of*

4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003). Volume 2893 of Lecture Notes in Computer Science (LNCS)., Paris/France, pages 236–247. Springer Verlag, 2003.

[SNLL05] Anjum Shehzad, Hung Quoc Ngo, S. Y. Lee, and Young-Koo Lee. A comprehensive middleware architecture for context-aware ubiquitous computing systems. In *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science*, 2005.

[SRA06] Maria A. Strimpakou, Ioanna G. Roussaki, and Miltiades E. Anagnostou. A context ontology for pervasive service provision. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, 2006.

[SRVN10] Sabine Schneider, Francesco Ricci, Adriano Venturini, and Elena Not. Usability guidelines for wap-based travel planning tools. In Ulrike Gretzel, Rob Law, and Matthias Fuchs, editors, *In Proceedings of the Information and Communication Technologies in Tourism 2010*, pages 125 – 136, 2010.

[Str03] Thomas Strang. *Service-Interoperabilität in Ubiquitous Computing Umgebungen.* PhD thesis, TU München, 2003.

[SZ06] Heshan Sun and Ping Zhang. The role of moderating factors in user technology acceptance. *International Journal of Human-Computer Studies*, 64:53 – 78, 2006.

[Tic98] Walter F. Tichy. Should computer scientists experiment more? *Cybersquare*, pages 32 – 40, 1998.

[TSSP08] Carlos A. Toro, César Sanín, Edward Szczerbicki, and Jorge Posada. Reflexive ontologies: Enhancing ontologies with self-contained queries. *Cybernetics and Systems*, 39:2:171 – 189, 2008.

[UG02] Michael Uschold and Michael Grüninger. Creating semantically integrated communities on the world wide web. In *in Proceedings of the Semantic Web Conference*, 2002.

[UNW08a] UNWTO.
International recommendations for tourism statistics 2008. Technical Report ST/ESA/STAT/SER.M/83/Rev.1, United Nations World Tourism Organization, 2008. ISBN 978-92-1-161521-0.

[UNW08b] UNWTO. Tourism satellite account: Recommended methodological framework 2008. Technical Report ST/ESA/STAT/SER.F/80/Rev.1, United Nations World Tourism Organization, 2008. ISBN 978-92-1-161520-3.

[Usc96a] Michael Uschold. Ontologies: Prindiples, methods and applications. *Knowledge Engineering Review*, 11:93 – 155, 1996.

[Usc96b] Mike Uschold. Building ontologies: Towards a unified methodology. In *In Watson I (ed) 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, 1996.

[vDHS07] Céline van Damme, Martin Hepp, and Katharina Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *In Proceedings of the ESWC Workshop "Bridging the gap between Semantic Web and Web 2.0"*, 2007.

[VFC05] David Vallet, Miriam Fernández, and Pablo Castells. An ontology-based information retrieval model. In *Proceedings of the European Semantic Web Conference*, 2005.

[VG07] Juan-Ignacio Vázquez-Gómez. *A Behavioural Model for Context-Aware Semantic Devices.* PhD thesis, Universidad de Deusto, January 2007.

[VMDD03] Viswanath Venkatesh, Michael G. Morris, Gordon D. Davis, and Fred D. Davis. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3):425 – 478, September 2003.

[Vos04] Jakob Voss. Begriffssysteme - ein vergleich verschiedener arten von begriffssystemen und entwurf des integrierenden datenmodells. Master's thesis, Humboldt Universität Berlin, 2004.

[VP00] Richard W. De Vaul and Alex Pentland. The ektara architecture: The right framework for context-aware wearable and ubiquitous computing applications. Technical report, The Media Laboratory, Massachusetts Institute of Technology, 2000.

[VSS+07] Claudia Villalonga, Martin Strohbach, Niels Snoeck, Michael Sutterer, Mariano Belaunde, Ern Kovacs, Anna V. Zhdanova, Laurent Walter Goix, and Olaf Droegehorn. Mobile ontology: Towards standardized semantic model for the mobile domain. In *In Proceedings of the first international Workshop on Telecom Service Oriented Architectures*, 2007.

[W3C04a] W3C. Composite capability/preference profiles (cc/pp): Structure and vocabularies 1.0. Technical report, W3C, 2004.

[W3C04b] W3C. Owl. Technical report, W3C, 2004.

[W3C04c] W3C. Rdf-schema. Technical report, W3C, 2004. Retrieved on March 28 2009.

[W3C08] W3C. Delivery context ontology. Technical report, W3C, 2008.

[Wan92] R. Want. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[Wat86] Donald A. Waterman. *A Guide to Expert Systems*. Addison-Wesley, 1986.

[Wei99] Mark Weiser. The computer for the 21st centuy. *American Scientific*, 3(3):pp. 3 – 11, September 1999. ISSN:1559-1662.

[Wer03] Hannes Werthner. Intelligent systems in travel and tourism. In *Proceedings of the 18th International joint conference on artificial intelligence (IJCAI-03)*, August 2003.

[WZGP04] Xiao-Hang Wang, D.Q. Zhang, T. Gu, and H.K. Pung. Ontology based context modelling and reasoning using owl. In *2nd IEEE Int. Conference on Pervasive Computing and Communications*, pages 18 – 22, 2004.

[YK07] Cheolho Yoon and Sanghoon Kim. Convenience and tam in a ubiquitous computing environment: The case of wireless lan. *Electronic Research and Applications*, 6:102 – 112, 2007.