Natalia A. Díaz Rodríguez

Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces

A Case Study on Ambient Assisted Living

Turku Centre for Computer Science

TUCS Graduate Programme and
UGR Official PhD Program on Information and Communication Technologies

TUCS Dissertations
No 186, April 2015

Editorial: Universidad de Granada. Tesis Doctorales
Auora: Natalia A. Díaz Rodríguez
ISBN: 978-84-9125-108-8
URI: http://hdl.handle.net/10481/40194

Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A case study on Ambient Assisted Living.

Natalia A. Díaz Rodríguez

To be presented, with the permission of the Department of Information Technologies at Åbo Akademi University, for public criticism in Auditorium Gamma, at ICT building, Turku Finland, on April 24th, 2015, at 12.00 noon. Joukahaisenkatu 3-5 A, Turku, Finland.

This doctoral thesis is presented within the official posgraduate program "Ciencias de la Computación y Tecnología Informática", the official posgraduate Master "Soft Computing and Intelligent Systems", and Turku Centre for Computer Science (TUCS) Graduate School.

Co-supervised Double European PhD Degree among:

Åbo Akademi University
Department of Information Technologies
Joukahaisenkatu 3-5 A, 20520 Turku, Finland

University of Granada
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Department of Computer Science and Artificial Intelligence
Periodista Daniel Saucedo Aranda, s/n, Granada, 18071, Spain

2015

Supervisors

Prof. Johan Lilius
Department of Information Technologies.
Embedded Systems Laboratory.
Åbo Akademi University
Joukahaisenkatu 3-5 20520 Turku (Finland)

Prof. Miguel Delgado Calvo-Flores
Department of Computer Science and Artificial Intelligence.
University of Granada,
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Periodista Daniel Saucedo Aranda s/n 18071 Granada (Spain)

Advisor

Assoc. Prof. Manuel Pegalajar Cuéllar
Department of Computer Science and Artificial Intelligence.
University of Granada,
Centro de Investigación en Tecnologías de La Información y de las Comunicaciones (CITIC-UGR)
Periodista Rafael Gómez, nº 2, 18071 Granada (Spain)

Reviewers

Assoc. Prof. Umberto Straccia
ISTI-CNR
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo".
Area della Ricerca CNR
Via G. Moruzzi 1, 56124 Pisa, Italy

Assoc. Prof. Pascal Hitzler
Wright State University.
Department of Computer Science and Engineering.
College of Engineering and Computer Science.
DaSe Lab for Data Semantics.
3640 Colonel Glenn Hwy Dayton, Ohio 45435 USA

Opponent

Prof. Gianluca Bontempi
Machine Learning Group
Computer Science Department
Université Libre de Bruxelles,
Bld du Triomphe CP 212 1050 Brussels, Belgium

ISBN 978-952-12-3139-1
ISSN 1239-1883

Abstract

Human activity recognition in everyday environments is a critical, but challenging task in Ambient Intelligence applications to achieve proper Ambient Assisted Living, and key challenges still remain to be dealt with to realize robust methods. One of the major limitations of the Ambient Intelligence systems today is the lack of semantic models of those activities on the environment, so that the system can recognize the specific activity being performed by the user(s) and act accordingly. In this context, this thesis addresses the general problem of knowledge representation in Smart Spaces. The main objective is to develop knowledge-based models, equipped with semantics to learn, infer and monitor human behaviours in Smart Spaces. Moreover, it is easy to recognize that some aspects of this problem have a high degree of uncertainty, and therefore, the developed models must be equipped with mechanisms to manage this type of information.

A fuzzy ontology and a semantic hybrid system are presented to allow modelling and recognition of a set of complex real-life scenarios where vagueness and uncertainty are inherent to the human nature of the users that perform it. The handling of uncertain, incomplete and vague data (i.e., missing sensor readings and activity execution variations, since human behaviour is non-deterministic) is approached for the first time through a fuzzy ontology validated on real-time settings within a hybrid data-driven and knowledge-based architecture. The semantics of activities, sub-activities and real-time object interaction are taken into consideration. The proposed framework consists of two main modules: the low-level sub-activity recognizer and the high-level activity recognizer. The first module detects sub-activities (i.e., actions or basic activities) that take input data directly from a depth sensor (Kinect). The main contribution of this thesis tackles the second component of the hybrid system, which lays on top of the previous one, in a superior level of abstraction, and acquires the input data from the first module's output, and executes ontological inference to provide users, activities and their influence in the environment, with semantics. This component is thus knowledge-based, and a fuzzy ontology was designed to model the high-level activities. Since activity recognition requires context-awareness and the ability to discriminate among activities in different environments, the semantic framework allows for modelling common-sense knowledge in the form of a rule-based system that supports expressions close to natural language in the form of fuzzy linguistic labels. The framework advantages have been evaluated with a challenging and new public dataset, CAD-120, achieving an accuracy of 90.1% and 91.1% respectively for low and high-level activities. This entails an improvement over both, entirely data-driven approaches, and merely ontology-based approaches.

As an added value, for the system to be sufficiently simple and flexible to be managed by non-expert users, and thus, facilitate the transfer of research to industry, a development framework composed by a programming toolbox, a hybrid crisp and fuzzy architecture, and graphical models to represent and configure human behaviour in Smart Spaces, were developed in order to provide the framework with more usability in the final application. As a result, human behaviour recognition can help assisting people with special needs such as in healthcare, independent elderly living, in remote rehabilitation monitoring, industrial process guideline control, and many other cases. This thesis shows use cases in these areas.

Sumario en Español

El reconocimiento de actividades humanas en entornos cotidianos es una tarea desafiante pero crítica en aplicaciones de Inteligencia Ambiental para poder lograr una adecuada asistencia ambiental (Ambient Assisted Living), y aún existen problemas clave por abordar para conseguir métodos más robustos. Por un lado, existe un amplio abanico de métodos basados en datos (por ejemplo, estadísticos o probabilísticos), que requieren el uso intensivo de datos etiquetados con el fin de entrenar el sistema para que aprenda las actividades. Por otro lado, disponemos de otra rama de métodos basados en el conocimiento (por ejemplo, semánticos o basados en ontologías), que requieren mecanismos versátiles para representar conocimiento y modelos para lograr razonamiento automático.

Mientras los enfoques basados en datos para el reconocimiento de actividades sufren de modelos estáticos *ad-hoc*, de escasez de datos, y de difícil escalabilidad [48], los modelos semánticos pueden satisfacer las necesidades de entornos basados en actividades personalizadas dependiendo del contexto, donde constantemente se están introduciendo nuevas tecnologías de sensores multimodales. Los modelos de datos simples se pueden conseguir a través de modelos de clave-valor y de etiquetado, mientras que dominios más complejos requieren formalismos más sofisticados, como los modelos basados en roles de objeto, modelos espaciales de contexto, u ontologías. En general, los requisitos para expresar comportamiento humano y el ambiente incluyen la capacidad para representar estructuras jerárquicas, relaciones complejas entre instancias de contexto y definiciones complejas basadas en otras más simples. Normalmente se requiere el uso de restricciones espaciales o temporales. Las ontologías han demostrado en la literatura ser una de las herramientas más prometedoras para lograr estos objetivos. Las ontologías pertenecen al paradigma de la *Web Semántica*, que originó como una colaboración del W3C y otros, para proporcionar un estándar para la definición de datos en la Web. La Web Semántica fue definida [29] por Berners-Lee et al. en el 2001 como "una extensión de la Web actual en la que se provee a la información con un significado bien definido, lo que permite a personas y computadores trabajar en cooperación". En la Web Semántica, las ontologías representan la principal tecnología para crear interoperabilidad a nivel semántico. Esto se logra mediante la creación de una ilustración formal de los datos, haciendo posible el compartir y reutilizar la ontología en toda la Web.

Un enfoque combinado para reconocer actividades es utilizando ambos paradigmas. Los enfoques híbridos requieren tanto de sentido común o conocimiento experto para modelos basados en el conocimiento, como de un modelado robusto basado en datos que también pueda capturar las variaciones de la actividad y la incertidumbre. El principal objetivo de esta tesis ha sido proponer un sistema efectivo en esta categoría.

Una de las principales limitaciones de los sistemas de Inteligencia Ambiental en la actualidad es la falta de modelos semánticos de actividades y del medio ambiente, para que el sistema pueda reconocer la actividad específica que se lleva a cabo por los individuos, y así actuar en consecuencia. En este contexto, esta tesis aborda el problema general de la representación del conocimiento en espacios inteligentes, teniendo como objetivo principal el desarrollo de modelos basados en el conocimiento, equipados con semántica, para aprender, inferir y controlar los comportamientos humanos en espacios inteligentes. Además, es fácil de reconocer que algunos aspectos de este problema tienen un alto grado de incertidumbre. Por ejemplo, a menudo faltan lecturas de sensores o hay variaciones en la ejecución de las actividades, que bien los humanos realizan de manera diferente, o de manera no determinista. Al mismo tiempo, los usuarios utilizan (diferentes) objetos de manera diferente. En consecuencia, se deben soportar variaciones en el comportamiento, ya que las actividades pueden llevarse a cabo de manera diferente o en diferente orden. Por tanto, los modelos desarrollados deben estar equipados con mecanismos para manejar este tipo de información imprecisa.

Para que el sistema sea lo suficientemente sencillo y flexible para poder ser gestionado por usuarios no expertos, y así, poder facilitar la transferencia de conocimiento e investigación a la industria, se ha desarrollado una librería para programar el espacio inteligente, una arquitectura híbrida crisp y difusa, así como modelos gráficos para representar y configurar reglas sobre comportamientos humanos en el espacio inteligente, con el fin de proporcionar al sistema más usabilidad en la aplicación final.

Una vez propuestos los componentes arquitectónicos necesarios del sistema, con el fin de permitir el modelado y el reconocimiento de un conjunto de comportamientos complejos de la vida real (donde la vaguedad y la incertidumbre son inherentes a la naturaleza humana de los usuarios que los realizan), construímos una ontología difusa y un sistema semántico híbrido.

El sistema propuesto se compone de dos módulos principales: el reconocedor de sub-actividades de bajo nivel, y el reconocedor de actividades de alto nivel. El primer módulo detecta sub-actividades, es decir, acciones o actividades básicas que toman datos de entrada directamente del sensor de profundidad Kinect. En nuestra aplicación, *Dynamic Time Warping* [57], que se caracteriza por ser un método basado en datos o de machine learning, se utiliza para aprender y reconocer estas sub-actividades de bajo nivel. La principal aportación de esta tesis aborda el segundo componente del sistema híbrido, que se basa en la parte anterior, a un nivel superior de abstracción, y obtiene los datos de entrada de la salida del primer módulo. Este módulo ejecuta inferencia ontológica para proporcionar semántica a usuarios, actividades y su influencia en el medio ambiente. Este componente, por tanto, se basa en el conocimiento, y utiliza la ontología difusa diseñada para modelar actividades de alto nivel. Como el reconocimiento de actividades depende del contexto y de la capacidad de poder discriminar entre actividades en diferentes entornos y configuraciones, el sistema semántico permite modelar conocimiento de sentido común en forma de un sistema basado en reglas que soporta expresiones cercanas al lenguaje natural con etiquetas lingüísticas difusas.

Para poder realizar el seguimiento de sub-actividades, así como su reconocimiento a partir de datos de vídeo de profundidad, además de para lograr un modelo más modularizado que permita que la flexibilidad sea parte del proceso de reconocimiento, se ha proporcionado significado formal a diferentes unidades de contexto en la ontología.

Las ventajas del sistema híbrido se han validado con un nuevo y desafiante conjunto de datos público, CAD-120 (Cornell Activity Dataset) [128], con 10 actividades en el entorno del hogar, realizadas por 4 usuarios. El sistema propuesto obtiene mejoras estadísticamente significativas en cuanto a la tasa de aciertos, precisión, y exhaustividad. Para la primera etapa basada en datos del sistema (de reconocimiento de sub-actividades), estos valores fueron 91,5, 97 y 90,1%, respectivamente, mientras que para la última etapa ontológica basada en el conocimiento (del sistema de reconocimiento de actividades de alto nivel), se logran un 84,1% de tasa de aciertos, 97,4% de precisión y 82,9% de exhaustividad. Por otro lado, si se asume un escenario ideal con una tasa de aciertos del 100% en la clasificación de las sub-actividades de entrada (es decir, suponiendo que todas las sub-actividades se reconozcan adecuadamente en la primera fase), se logra una tasa de aciertos del 90,8%, precisión de 98,1% y una exhaustividad del 91,07%.

El tratamiento de datos inciertos, incompletos o vagamente expresados es abordado por primera vez a través de una ontología difusa, y validado en situaciones en tiempo real con un conjunto de datos externo. Esto supone una mejora con respecto a ambos enfoques existentes en el estado del arte, es decir, tanto enfoques totalmente basadas en datos, como metodologías que meramente utilizan ontologías.

Las contribuciones de la tesis se pueden resumir en:

1. Un estado del arte sobre metodologías y enfoques para el reconocimiento de comportamientos humanos, y un análisis de ontologías existentes para el

mismo propósito.

2. Un conjunto de componentes de infraestructura, tales como una arquitectura de razonamiento híbrida crisp-difusa, un módulo de programación semántico, y un modelo de lenguaje visual para el usuario final que: a) permite la programación de aplicaciones personalizadas por el usuario con simples reglas "SI-ENTONCES", b) no requiere conocimientos de programación, Web Semántica ni lógica difusa, c) se basa en triples y grafos para preservar el modelo semántico de RDF, d) incluye una arquitectura de publicación/suscripción (Smart-M3) para evitar consultas constantemente, y e) soporta el modelado e inferencia con conocimiento impreciso.

3. Una ontología difusa que permite modelar acciones, actividades, comportamientos, ubicaciones, tiempo, diferentes tipos de usuarios (individuales, grupos) y la incertidumbre inherente al contexto.

4. Un sistema híbrido que combina el reconocimiento de actividades por visión por computador con modelos semánticos que a) mejora la sensibilidad al contexto en entornos dinámicos, b) mejora la tasa de aciertos, precisión, exhaustividad y la interpretabilidad y expresividad del modelo (de manera más cercana al lenguaje natural), c) evita la necesidad de entrenar el sistema de nuevo cuando se introducen nuevas actividades de alto nivel para ser reconocidas en el sistema.

5. Un modelado difuso de actividades humanas más robusto por medio del tratamiento de datos imprecisos, expresados con vaguedad, incompletos o inciertos. El modelado difuso, al mismo tiempo, permite relajar el modelo y facilita su flexibilidad.

Como resultado de todas estas contribuciones, el reconocimiento de actividades humanas puede ser herramienta clave para conseguir una mejor asistencia a personas con necesidades especiales, mayores que viven independientemente, así como en asistencia sanitaria, o en monitorización o rehabilitación remota, y en control de procesos o protocolos industriales, entre otros casos. En esta tesis se muestran ejemplos de uso en algunas de estas áreas. Trabajos futuros deben centrarse en los efectos derivados del punto 4c, es decir, tratar de reducir el trabajo manual requerido para extraer y representar conocimiento experto, el cual es considerable actualmente, con el fin de representar las reglas de dominio de manera suficientemente coherente, específica, general y reproducible.

Svenska Sammanfattning

Att känna igen mänsklig aktivitet i vardagliga miljöer är en viktig men utmanande uppgift för applikationer skapade för intelligenta miljöer (eng. "Ambient Intelligence"). Det här måste uppnås för att skapa ett fungerande boende med IT-stöd (eng. Ambient Assisted Living) men centrala utmaningar återstår fortfarande för att fullständigt uppnå tillräckligt robusta metoder. Å ena sidan finns det ett brett område av så kallade datadrivna metoder (t.ex. statistiskt eller sannolikhetsbaserade), som kräver intensiv användning av märkta data för att lära sig om aktiviteter, det vill säga, träna systemet. Å andra sidan så finns det en uppsjö av kunskapsbaserade metoder (t.ex. semantiskt eller ontologibaserade) som kräver mångsidiga mekanismer för att representera kunskap och modellera regler för att utföra automatiska resonemang.

Medan datadrivna metoder för igenkännande av aktiviteter lider av *ad-hoc* statiska modeller, brist på data och skalbarhet [48], så kan semantiska modeller uppfylla behoven hos kontextmedvetna personliga aktivitetsbaserade miljöer, där multimodal sensorteknik konstant introduceras. Enkla datorsystem kan modelleras med hjälp av nyckelvärden och markeringsmodeller (eng. Markup Model), medan mer komplicerade domäner kräver mer sofistikerade formalismer såsom objekt-roll baserade modeller, kontext baserade rumsliga modeller eller ontologier. Generellt så är kravet på uttrycksfullhet för mänskligt beteende och miljörepresentation att de ska inkludera möjligheten att representera hierarkiska strukturer, komplexa relationer mellan kontext-instanser och komplexa definitioner baserade på enklare aktiviteter. Oftast, kan restriktioner, som kan vara rumsliga eller tidsmässiga, hjälp att modellera. Forskning har visat att ontologier kan vara ett av de mest lovande verktygen för att uppnå dessa mål. Ontologier hör till den semantiska webben, som introducerades av bland annat W3C (World Wide Web Consortium) med syfte att skapa en standard för att definiera data på webben. Den semantiska webben definierades av Tim Berners-Lee et al. i 2001 [29] som "en förlängning av den nuvarande webben där information ges väldefinierad betydelse, för att bättre möjliggöra samarbete mellan datorer och människor". I den semantiska webben så utgör ontologier den viktigaste tekniken för att skapa interoperabilitet på en semantisk

nivå. Detta uppnås genom att skapa en formell illustration av data, vilket gör det möjligt att dela och återanvända ontologin över hela webben.

En kombinerad metod för igenkänning av aktivitet använder sig av båda paradigmerna. Hybrida tillvägagångssätt kräver både förnuftiga kunskapsbaserade modeller och robusta datadrivna modeller som också kan fånga variationerna och osäkerheten i aktiviteten. Denna avhandling kommer avslutningsvis att föreslå ett nytt system i denna kategori.

En av de stora begränsningarna gällande system för intelligenta miljöer är för tillfället bristen på semantiska modeller för miljö-baserade aktivitet, vilka skulle möjliggöra för systemet att känna igen den specifika aktivitet som utförs av användarna och agera därefter. Denna avhandling behandlar det allmänna problemet med kunskapsrepresentation i smarta utrymmen (eng. Smart Spaces) i denna kontext. Huvudsyftet är att utveckla kunskapsbaserade modeller, som kan hantera semantik för att lära, dra slutsatser och övervaka mänskliga beteenden i smarta utrymmen. Det lätt att inse att vissa aspekter av detta problem har en hög grad av osäkerhet. Till exempel förekommer det ofta att det saknas sensoravläsningar, variationer på människor som utför aktiviteter på olika sätt eller på ett ickedeterministiskt sätt. Samtidigt så ökas problematiken då individuella användare använder (olika) objekt på olika sätt. Följaktligen bör beteendeförändringar vara med i beräkningarna, eftersom aktiviteter kan utföras på olika sätt eller i olika ordning. Därför måste de utvecklade modellerna vara utrustade med mekanismer för att hantera denna typ av oprecis information.

För att systemet ska vara tillräckligt enkelt och flexibelt för att kunna hanteras av ickeexpertanvändare, och därmed underlätta överföringen av forskningen till industrin, så har ett ramverk bestående av en programmeringsverktygslåda, en hybrid skarp och oskarp arkitektur och grafiska modeller för att representera och konfigurera mänskliga beteende-regler i smarta utrymmen utvecklats. Det ger ramverket med större användbarhet i den slutliga tillämpningen.

Efter att kraven på de arkitektoniska komponenterna fastslagits så byggdes en oskarp ontologi och ett semantiskt hybridsystem för att möjliggöra modellering och igenkännande av en rad komplexa verkliga situationer (där vaghet och osäkerhet tillhör den mänskliga naturen hos de användare som utför det).

Det föreslagna ramverket består av två huvudmoduler: en låg-nivå underaktivitets igenkännare och en hög-nivå aktivitets igenkännare. Den första modulen upptäcker underaktiviteter (dvs. åtgärder eller grundläggande aktiviteter) som tar indata direkt från en djupsensor (Kinect). I vår tillämpning applicerades dynamisk tidsvarpning (eng. *Dynamic Time Warping* [57]) för att lära och känna igen dessa låg-nivå underaktiviteter, som fungerar som en maskininlärningsbaserad datadriven metod. Det viktigaste resultatet av denna avhandling behandlar den andra komponenten i hybridsystemet, som verkar ovanpå den tidigare, i en högre abstraktionsnivå, där indata fås från den första modulens utsignal. Denna modul utför ontologiska slutledningar för att tillhandahålla semantik åt användare, aktiviteter och deras påverkan på miljön. Denna komponent är alltså kunskapsbaserad och använder den oskarpa ontologi som konstruerats för att modellera hög-nivå aktiviteter. Eftersom erkännande av aktiviteter kräver kontextmedvetenhet och förmågan att skilja på aktiviteter i olika miljöer och situationer, så kan det semantiska ramverket modellera kunskap baserad på sunt förnuft i form av ett regelbaserat system som stödjer uttryck som är nära naturligt språk, detta görs möjligt med hjälp av oskarpa lingvistiska etiketter.

Betydelsen av kontext kartlades för att bättre kunna utföra underaktivitetsspårning och igenkännande från videodata med djup och för att uppnå en mer löst kopplad modell som låter flexibilitet vara en del av igenkänningsprocessen.

Fördelarna med det hybrida ramverket validerades med hjälp av en utmanande och ny offentlig datamängd, CAD-120 (Cornell Activity Dataset) [128], som innehåller 10 aktiviteter i hemmiljö som utförs av 4 olika användare. Statistiskt signifikanta förbättringar erhölls beträffande precision, återkallelse och noggrannhet. För det första datadrivna steget av AR-systemet (underaktivitet igenkännande), så var värden 91,5, 97 och 90,1 %, respektive, medan värdena för den kunskapsbaserade, ontologiska och sista etappen av AR systemet (på hög-nivå) var: 84,1 % precision, 97,4 % återkallelse och 82,9 % noggrannhet. Däremot, om ett perfekt scenario med 100 % noggrannhet på märkta ingångs delaktiviteter utförs (med antagandet att alla underaktiviteter är korrekt redovisade i den första fasen), så uppnås en precision på 90,8 %, en återkallelse av 98,1 % och en noggrannhet på 91,07 %.

Hanteringen av osäkra, ofullständiga och vaga uppgifter behandlas för första gången med hjälp av en oskarp ontologi och valideras i realtid för olika situationer genom att använda en offentlig djup-video baserat datamängd. Detta innebär en förbättring jämfört med både helt datadrivna metoder och enbart ontologibaserade metoder som presenteras mera ingående i *state-of-the-art* sektionen.

Resultaten från avhandlingen kan således sammanfattas som:
1. En sammanfattning av toppmoderna metoder och strategier för mänskligt beteende igenkänning, samt en analys av ontologiers användbarhet för dessa ändamål.
2. En uppsättning infrastrukturkomponenter, såsom en hybrid skarp-oskarp arkitektur för resonemang, en semantisk modul för utveckling och en visuell språkmodell för slutanvändare som: a) möjliggör programmering av applikationer anpassade efter användaren, med enkla OM-DÅ regler, b) inte kräver kunskap om varken programmering, den semantiska webben eller oskarp logik, c) är grafbaserad för att bevara den RDF-baserade semantiska

modellen, d) innehåller en publicera/prenumerera arkitektur (Smart-M3) för att undvika återkommande förfrågningar och e) stödjer modellering och resonemang med oprecis kunskap.
3. En oskarp ontologi för att modellera åtgärder, aktiviteter, beteenden, platser, tid, olika typer av användare (enskilda, grupper) samt den naturliga osäkerheten knuten till kontexten.
4. Ett hybrid igenkänningssystem för aktiviter som kombinerar datorseende med semantiska modeller som a) förbättrar kontext-medvetenhet i dynamiska miljöer, b) förbättrar noggrannhet, precision, återkallelse Och tolkningsbarheten av modellen och dess uttryckbarhet (mera specifikt ett naturligt språk), c) undviker omskolning av nya hög-nivå aktiviteter.
5. En mer robust, oskarp modellering av mänsklig aktivitet som tar itu med oprecisa, vaga och osäkra uppgifter och på samma gång, underlättar glapp och flexibilitet i modellen.

Som ett resultat av alla dessa bidrag, kan igenkänning av mänskligt beteende bättre underlätta stödet till människor med särskilda behov, såsom inom hälsovården, självständigt äldreboende, fjärrövervakad rehabilitering, industriell reglering av riktlinjer för processer och många andra fall. Denna avhandling presenterar användningsfall för vissa av dessa områden. Framtida forskning bör koncentrera på negativa effekter av punkt 4c, dvs. minska det omfattande manuella arbete som för tillfället krävs av experter för att representera tillräckligt konsekventa, specifika och allmänna domänregler.

Acknowledgements

This PhD thesis has been the fruit of four intense years of research work and different projects. The thesis would have not become what it is without a large amount of people involved and that I am grateful to.

First of all, I have to thank my supervisors. In first place, I would like to thank Prof. Johan Lilius for counting on me to start at the Embedded Systems Laboratory after being Erasmus student at Åbo Akademi in Turku, Finland. I was very lucky to get to experience Finnish efficiency, flexibility and smoothness provided by Prof. Lilius, who always is supportive, motivating and allowed me to enjoy the best part of the PhD by attending great conferences and the most inspiring and productive summer schools, where I have met incredible people and acquired not only fruitful colleagues, but also friends for life (Sumi, Martin, Christos, Daniel, Andrei-Adnan, Christina, Larissa, Kamalika, Tanwi, Lorenzo, Enrico and Alireza, Joao, Serdar, Alex, Michèle, Bruno, Jens, Fernando, Jacqueline, Pascal, Nekane, Georgios and Ulli, Stefano, etc.). I admire my main supervisor for his efficacy getting things solved and forward, as well as his excitement for getting new projects or gadgets for the lab for all to play around.

On the other side, I got Spanish fully devoted support and advice from my Spanish supervisor, Prof. Miguel Delgado Calvo-Flores, from years back (starting with a collaboration grant back in 2009). His warmth and wisdom are always comforting and encouraging. I am totally grateful as well to my advisor, Assoc. Prof. Manuel Pegalajar Cuéllar, who was key in achieving what the thesis is today. His always fast replies and unconditional support, critics and guidance have had a great positive impact on my work and have taught me to do quality work.

Every time I had a meeting with any of my (in practice) three supervisors, all PhD fears and worries were smoothly solved, and an uplifting mood was immediately propagated in me. I could not have been luckier having greater supervisors. Thanks to all of you for having guided me throughout all these years, I could not be happier nor feel more privileged and thankful to you. I really hope we can continue collaborating for life.

I am grateful to Assoc. Prof. Umberto Straccia and Assoc. Prof. Pascal Hitzler, who kindly agreed to serve as reviewers of the thesis, and provided me with comments and motivation crucial to improve the thesis, as well as interesting insights for future directions. I am as well greatly grateful to Prof. Gianluca Bontempi, for accepting acting as opponent of my thesis.

Without my co-authors, the articles would not have been possible. I want to thank Manuel Pegalajar Cuéllar, Olmo León Cadahía, Mohsin Saleemi, Shohreh Hosseinzadeh, Seppo Virtanen, Stefan Grönroos, Franck Wickström, Petteri Karvinen, Anders Berg, Marion Karppi, Espen Suenson, Ivan Porres, Pasi Kankaanpää, Riitta Danielsson-Ojala, Hanna Pirinen, Lotta Kauhanen, Sanna Salanterä, Sebu Björklund, Joachim Majors, Kimmo Rautanen, Tapio Salakoski, Ilona Tuominen and Nauman Khan. Apart from co-author, I also thank Robin Wikström and Wictor Lund, for being always available to help, as well as to proofread the thesis' Swedish summary.

Regarding colleagues, first of all I would like to thank the Åbo Akademi gang, where I spent 5 years in total both as Erasmus student, research assistant and PhD student. Thanks to the "embedded" gang and IT-Dept. colleagues: Ersfolk, Andreas, Stefan, Teddy, Fareed, Sudeep, Fredric, Simon, Wictor, Robert, Dag, Ranjita, George, Jerker, Hannu, Petteri, Anders, Coralie, Billy, Marat, Leonidas, Björn, Inna, Yuliya, Linas, Elena, Juuso, Samuel, Peter and Bea.

Additionally in the ÅA side I would like to thank Annamari, Åke, Ragnar and Niclas for which it was fun to work as teaching assistant, as well as colleagues like Linda, Mats, Sébastien, Marta, Miki, Nazrul, etc. for always supporting and be willing to collaborate and get excited about things like me. I also enjoyed the SEMPRE discussions among the 3 department labs, and organizing the Department monthly breakfast talks with Matteo, Sepinoud and Henrik. I thank Johan, Ivan Porres and Ion Petre for his encouragement and initiatives on increasing scientific curiosity and university outreach. Also thanks to Ralph-J. Back and Ivan for introducing me to the IT Department at the Gaudi Software Factory in summer 2008 after my Erasmus.

I also would like to thank Sanna Salanterä and Riitta Danielsson-Ojala's Nursing Sciences group at University of Turku for their enthusiastic group meetings and project ideas that we together developed for a Smart Hospital. I hope our collaboration is not over.

On the Spanish side, I would like to thank Fernando Bobillo for his always remote but patient and efficient support with *FuzzyDL* and *fuzzy OWL* tools. I also thank María Ros, Amparo Vila and Maria José Bautista for always being available to help.

I would also like to thank the "fuzzy coffee" colleagues (and party friends): Juanan, Manu, Sara, Jose Antonio, Dani, Rafa, Olmo, Irene, Raquel, Isaac, Rosa, Nacho, Alberto, Antonio, Diana, Fran, Pablo, Vicky, Luis, Sergio, Kasia, Weronika, Dragi, Karel, Oresti, Laila and Brahim. You made my

8 months in Granada fly and super fun with our language exchange meetings, etc. Likewise, old friends must be thanked for keeping true friendship and refreshing tapas breaks: Rosalía, Vito, Esther, Elsi, Marta, M. José, Miriam, Laura, M. Elena, Esther, M. Carmen, and all my dear *ETSIIT-anos* friends: Lucía, Marta, Irene, David x2, Marce, Juanga, Carlos x2, Alfonso, Javi, Manu, Andrés, Serchu, Alicia, Juanmi, etc.

As for the final slope of the thesis, I thank Aki, Paul, Alberto, Rim, Silvia, Vana, Jenny, Rohan, Shiva, Julià, Gabriele, Alejo, Borja, Luigi, Victor, Laura, Sandeep and rest of Philips and Eindhoven gang, for keeping my energy and mood levels under stress conditions above average ;).

Regarding administrative personnel, I have to thank Inmaculada Domínguez for her invaluable help getting the double degree co-supervision agreement formalized and implemented, as well as Christel Engblom, Tove Österroos, Tomi Mäntylä and Beate Krug for their administrative help in both Turku and Granada.

Doing PhD was my best decision ever, I have met the most awesome people I could ever meet, fun, passionate and hard working people with innovative ideas and ideals from which I have enriched myself not only professionally but they have made me grow greatly at a personal level. Communities such as Boost Turku, the rowing club, the Indian gang or the cheese club made my stay in Turku unforgettable. I thank Matteo, Hannele, Daniela, Michelle, Ida, Giacomo, Luca, Martina, Jenni, Céline, Bogdan, Dasha, Erla, Joanna, Diana, Charmi, Faezeh, Irum, Joanna, Gloria, Soile, Mohit, Shishir, Senthil, Hari, Pramod, Saara, Finnish sister Maaria, Hanna, Aino x2, Laura x2, Linnéa, Jocke, Gautam, Jose, Tingting, Sebastian, Borna, etc. Friends like you are hard to find. Some of you, more than friends, were to me a Finnish family, e.g., Pirjo and Göran, Olle, Elina. Micke, you always motivated me to stay positive and taught me that nothing is impossible, you influenced my thinking by spreading humanitarian and science curiosity. Pasi, you always see life positively, fun, and it is optimistic and refreshing to see you. Helen, thanks for being a wise and enthusiastic sister. I am very grateful to have now two homes and to know what *sisu* means.

Finally, last, but not least, I heartily thank my parents, siblings and grandma for always supporting me and stand the distance for this many years. You always provided me love, wisdom and good advice for my career to progress in the right direction. I love you so much *shikis*!

[Tras agradecer a los contactos del entorno académico, no puedo más que agradecer de corazón a mis padres, hermanos y abuela por siempre apoyarme y soportar la distancia durante todos estos años. Siempre me habeis provisto con amor, sabiduría y buenos consejos para que mi carrera avance en la dirección correcta. Os quiero mucho, shikis!]

To conclude, I thank TUCS (Turku Centre for Computer Science), Finnish Cultural Foundation, Hans Bang Foundation, Nokia Foundation and Google

Anita Borg Scholarship for the funding provided. Other foundations which provided support for conferences and schools were Heidelberg Laureate Forum Foundation, EU 7th Framework Program's Partnership for Advanced Computing in Europe Implementation Phase project (PRACE-3IP). I also thank EIT ICT Labs Doctoral School for its Doctoral Program on Innovation and Entrepreneurship studies and the inspiring opportunities this school created for me, as well as for providing me with the entrepreneurship mindset. Funding was also obtained from DIEM (Devices and Interoperability Ecosystems) project within the Tekes-TIVIT ICT-shok program founded by the Finnish Funding Agency for Innovation, the AMEBA (Agent Based Management of Embedded Data Reserves) project from Academy of Finland, and the Active Healthy Ageing (AHA) Project founded by EIT ICT Labs.

Natalia. Granada, January 2015.

Natalia Díaz Rodríguez, born 1986 in Granada (Spain)
Computer Engineering Master (2010)
Soft Computing and Intelligent Systems Master (2012)
Doctor of Embedded Systems, Computer Science and Artificial Intelligence (2015)

List of original publications

In descending order of publication:

1. *Handling real-world context-awareness, uncertainty and vagueness in real-time human activity tracking and recognition with a fuzzy ontology-based hybrid method.* N. Díaz Rodríguez, O. León Cadahía, M. P. Cuéllar, J. Lilius, M. Delgado-Calvo-Flores. MDPI Physical Sensors. 14 (10), pp. 18131-18171, 2014.

2. *A fuzzy ontology for semantic modelling and recognition of human behaviour.* Natalia Díaz Rodríguez, M.P. Cuéllar, Johan Lilius, Miguel Delgado Calvo-Flores. Knowledge-Based Systems, 66. pp. 46-60. 2014.

3. *A Survey on Ontologies for Human Behaviour Recognition.* Natalia Díaz Rodríguez, Manuel Pegalajar Cuéllar, Johan Lilius, Miguel Delgado Calvo-Flores. ACM Computing Surveys 46, 1–32, 2014.

4. *Understanding Movement and Interaction: an Ontology for Kinect-based 3D Depth Sensors.* Natalia Díaz Rodríguez, Robin Wikström, Johan Lilius, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores. In: José Bravo, Sergio Ochoa (Eds.), 7th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2013), Lecture Notes in Computer Science, 254-261, Springer, 2013.

5. *Rapid Prototyping of Semantic Applications in Smart Spaces with a Visual Rule Language.* Natalia Díaz Rodríguez, Johan Lilius, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores. In: Jonna Häkkila, Kamin Whitehouse, Antonio Krüger, Yoshito Tobe, Otmar Hilliges, Koji Yatani, Anind K. Dey, Hans-Werner Gellersen, Elaine M. Huang, Timo Ojala, Silvia Santini, Friedemann Mattern (Eds.), Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, UbiComp '13 Adjunct, 1335–1338, ACM, 2013.

6. *An Approach to Improve Semantics in Smart Spaces Using Reactive Fuzzy Rules.* Natalia Díaz Rodríguez, Johan Lilius, Manuel Pegalajar

Cuéllar, Miguel Delgado Calvo-Flores. In: Witold Pedrycz, Marek Z. Reformat (Eds.), IFSA World Congress - NAFIPS Annual Meeting (IFSA/NAFIPS), 436–441, IEEE, 2013.

7. *Extending Semantic Web Tools for Improving Smart Spaces Interoperability and Usability.* Natalia Díaz Rodríguez, Johan Lilius, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores. In: Sigeru Omatu, José Neves, Juan M. Corchado, Juan F. De Paz Santana, Sara Rodríguez González (Eds.). Distributed Computing and Artificial Intelligence, 10th International Conference, Advances in Intelligent Systems and Computing 217, 45–52, Springer , 2013.

8. *Ontology Driven Smart Space Application Development.* M. Mohsin Saleemi, Natalia Díaz Rodríguez, Espen Suenson, Johan Lilius, Ivan Porres. In: Salvatore F. Pileggi, Carlos Fernandez-Llatas (Eds.), Semantic Interoperability: Issues, Solutions, and Challenges, 101–125. River Publishers, 2012.

9. *A Framework for Context-Aware Applications for Smart Spaces.* M. Mohsin Saleemi, Natalia Díaz Rodríguez, Johan Lilius, Ivan Porres. In: Sergey Balandin, Yevgeni Koucheryavi, Honglin Hu (Eds.), Proceedings of The 4th conference on Smart Spaces ruSMART 2011, Lecture Notes in Computer Science, 14–25, Springer, 2011.

10. *Programming Biomedical Smart Space Applications with BioImageXD and PythonRules.* Natalia Díaz Rodríguez, Pasi Kankaanpää, M. Mohsin Saleemi, Johan Lilius, Ivan Porres. In: Adrian Paschke, Albert Burger, Paolo Romano, M. Scott Marshall, Andrea Splendiani (Eds.). 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences (SWAT4LS 2011), 10–11, ACM, 2011.

Other Publications:

11. *Registeted Nurses' Experiences with the Medication Administration Process.* Hanna Pirinen, Lotta Kauhanen, Riitta Danielsson-Ojala, Johan Lilius, Ilona Tuominen, Natalia Díaz Rodríguez, Sanna Salanterä. Submitted to Journal of Clinical Nursing (JCN), Wiley. 2014.

12. *Erratum to: Exploiting smart spaces for interactive TV applications development.* Saleemi, M. Mohsin, Díaz Rodríguez, N. and Lilius, J. (2014). The Journal of Supercomputing, pages 1-18. Springer.

13. *A semantic security framework and a context-aware role-based access control scheme for Smart Spaces.* Shohreh Hosseinzadeh, Natalia Díaz Rodríguez, Seppo Virtanen, Johan Lilius. Submitted.

14. *Can IT health-care applications improve the medication tray-filling process at hospital wards? An exploratory study using eye-tracking and stress response.* Natalia Díaz Rodríguez, Johan Lilius, Riitta Danielsson-Ojala, Hanna Pirinen, Lotta Kauhanen, Sanna Salanterä, Sebu Björklund, Joachim Majors, Kimmo Rautanen, Tapio Salakoski, Ilona Tuominen. Accepted in 16th International Conference on E-health Networking, Application & Services (IEEE HealthCom) 2014, Natal, Brazil.

15. *Smart Dosing: A mobile application for tracking the medication tray-filling and dispensation processes in hospital wards.* N. A. Khan, N. Díaz Rodríguez, R. Danielsson-Ojala, H. Pirinen, L. Kauhanen, S. Salanterä, J. Majors, S. Björklund, K. Rautanen, T. Salakoski, I. Tuominen, I. Porres and J. Lilius. WISHWell'14: 6th International Workshop on Intelligent Environments Supporting Healthcare and Well-being (Best paper award). Workshop within the European Conference on Ambient Intelligence, 2014, Eindhoven, The Netherlands.

16. *Semantic and Fuzzy Modelling of Human Behaviour in Smart Spaces. A case study on Ambient Assisted Living and Remote Rehabilitation.* Natalia Díaz Rodríguez, Manuel Pegalajar Cuéllar, Johan Lilius, Miguel Delgado Calvo-Flores. Feb 19-21. 2014. In Bailón-Moreno, R. (dir.), I Congreso Nacional Multidisciplinar de Jóvenes Investigadores, I CNMJI (The National Multidisciplinary Congress of Young Researchers). Granada (Spain). http://hdl.handle.net/10481/32082.

17. *Monitorizado remoto de ejercicios de rehabilitación en tiempo real con sensores de profundidad 3D y tecnologías semánticas.* Natalia Díaz Rodríguez, Robin Wikström, Stefan Grönroos, Franck Wickström, Petteri Karvinen, Anders Berg, Shohreh Hosseinzadeh, Johan Lilius, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores, Marion Karppi, Feb. 19-21. 2014. In Bailón-Moreno, R. (dir.), I Congreso Nacional Multidisciplinar de Jóvenes Investigadores, I CNMJI (The National Multidisciplinary Congress of Young Researchers). Granada (Spain). http://hdl.handle.net/10481/32082.

18. *M3 Interoperability for Remote Rehabilitation with Kinect.* Natalia Díaz Rodríguez, Stefan Grönroos, Franck Wickström, Petteri Karvinen, Anders Berg, Shohreh Hosseinzadeh, Marion Karppi, Johan Lilius. In: Juha-Pekka Soininen, Sergey Balandin, Johan Lilius, Petri Liuha, Tullio Salmon Cinotti (Eds.). Open International M3 Semantic Interoperability Workshop, 21, 153-163, TUCS Lecture Notes, 2013.

19. *A Scalable Distributed M3 Platform on a Low-Power Cluster.* Anders Berg, Petteri Karvinen, Stefan Grönroos, Frank Wickström, Natalia Díaz Rodríguez, Shohreh Hosseinzadeh, Johan Lilius, In: Juha-Pekka Soininen, Sergey Balandin, Johan Lilius, Petri Liuha, Tullio Salmon Cinotti (Eds.), Open International M3 Semantic Interoperability Workshop , TUCS Proceedings 21, 49– 58, TUCS, 2013.

20. *Towards Ontology-Driven Development of Ubiquitous and Intelligent interactive TV Applications.* M. M. Saleemi, N. Díaz Rodríguez, J. Lilius. In: Stephen S. Yau, Witold Pedrycz, Vincenzo Piuri (Eds.), The 10th IEEE International Conference on Pervasive, Intelligence and Computing (PiCom 2012), 1–8, IEEE, 2012.

Glossary

AAL Ambient Assisted Living: according to the European Commission the concept aims to prolong the time people can live in a decent way in their own home by increasing their autonomy and self-confidence the discharge of monotonously everyday activities to monitor and care for the elderly or ill person to enhance the security and to save resources. AAL refers to intelligent systems of assistance for a better healthier and safer life in the preferred living environment and covers concepts products and services that interlink and improve new technologies and the social environment[1].

ADL Activities of Daily Living: Routine activities that people tend do everyday without needing assistance. There are six basic ADLs: eating bathing dressing toileting transferring (walking) and continence[2].

AI Artificial Intelligence: the theory and development of computer systems able to perform tasks normally requiring human intelligence such as visual perception speech recognition decision-making and translation between languages.

AmI Ambient Intelligence: The term *Ambient Intelligence* (AmI) was introduced by the European Commission in 2001 [110] as a response to new user needs in ubiquitous environments. One of the latest definitions [22] describes AmI as *digital and proactive environments with capacity to sense the environment and assist users in their daily lives.* The major difference between *Ubiquitous Computing* and AmI is the introduction of AI in the latter. Thanks to that user-friendliness user-empowerment human assistance and easy interaction with efficient services are improved. Examples of AmI scenarios can be seen in [22, 175, 55].

AR or HAR (Human) Activity Recognition: Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions. Activity recognition is an important technology in pervasive computing because it can be applied to many real-life human-centric

[1] AALIANCE project – FP7/Cooperation/ICT.

[2] `http://www.investopedia.com/terms/a/adl.asp`

problems such as elder care and healthcare. Successful research has so far focused on recognizing simple human activities. Recognizing complex activities remains a challenging and active area of research.

DAML The DARPA Agent Markup Language (DAML) was the name of a US funding program at the US Defense Advanced Research Projects Agency (DARPA) started in 1999 by then-Program Manager James Hendler and later run by Murray Burke Mark Greaves and Michael Pagels. DAML is a markup language that is based on XML. DAML is designed to have a greater capacity than XML for describing objects and the relationships between objects to express semantics and to create a higher level of interoperability among Web sites. As the central research and development agency for the U. S. Department of Defense DARPA was instrumental in the creation of the Internet and many of its technologies. DARPA is developing DAML as a technology with intelligence built into the language through the behaviors of agents programs that can dynamically identify and comprehend sources of information and interact with other agents in an autonomous fashion[3].

DAML+OIL a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema and extends these languages with richer modelling primitives.

DIEM (Devices and Interoperability Ecosystems): a research-industry project within the Tekes-TIVIT ICT-shok program founded by Tekes the Finnish Funding Agency for Innovation (Jun. 2008- Dec. 2012) [4].

DL Description Logics: the most used languages to model formal ontologies. DL reasoning can support incremental progressive activity recognition and assistance as the activity unfolds. Ontology-based activity recognition provides a number of advantages [51]. In DL the terminology or the *TBox* is the vocabulary used for defining concepts and roles within a domain while all instances or named individuals conform assertions about a real world domain in the *ABox*. Statements in the TBox and ABox can be interpreted with rules and axioms in DL to enable reasoning and inference including satisfiability subsumption equivalence disjointness of classes classification consistency instance retrieval and realization [179]. DL reasoning supports decidability completeness and soundness in polynomial time complexity for an inexpressive DL and in exponential time complexity for expressive DLs [25].

DTW Dynamic Time Warping: in time series analysis DTW is an algorithm for measuring similarity between two temporal sequences which

[3] http://searchsoa.techtarget.com/definition/DAML

[4] DIEM Project: http://diem.fi/

may vary in time or speed.

fuzzyDL is a descriptive fuzzy description logics reasoner [31, 33] that allows to reason with fuzzy ontologies. It has been used in applications from matchmaking to fuzzy control.

GUI Graphical User Interface: a human-computer interface (i.e. a way for humans to interact with computers) that uses windows icons and menus and which can be manipulated by a mouse (and often to a limited extent by a keyboard as well).

HMM Hidden Markov Models: a finite set of states each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities.

IoT The Internet of Things is the network of physical objects accessed through the Internet as defined by technology analysts and visionaries. These objects contain embedded technology to interact with internal states or the external environment. In other words when objects can sense and communicate it changes how and where decisions are made and who makes them [5].

KB Knowledge Base: the underlying set of facts assumptions and rules which a computer system has available to solve a problem.

KP Knowledge Processor entity within M3 platform. They implement functionality and interact with the Smart Space by inserting/querying common information through the publish/subscribe SSAP protocol and also through SPARQL.

LDA Linear Discriminant Analysis: a classification method originally developed in 1936 by R. A. Fisher. It is simple mathematically robust and often produces models whose accuracy is as good as more complex methods. Its score function is to capture the notion of separability.

Ontology an ontology is a formal specification of a shared conceptualization [83]; a collection of predefined formal specification of terms that define concepts relationships and constraints within a domain. Ontologies can provide a class structure with constraints filling similar functions as a database schema.

OO-Programming Object Oriented Programming: a programming paradigm that represents the concept of "objects" that have data fields and associated procedures known as methods.

ORM Object-Role Modelling: a powerful method for designing and querying database models at the conceptual level where the application is described in terms easily understood by non-technical users.

OWL Web Ontology Language: a family of knowledge representation languages or ontology languages for authoring ontologies or knowledge

[5] IoT: `http://www.cisco.com/web/solutions/trends/iot/overview.html`

bases. The languages are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. OWL is based on the knowledge representation formalism of Description Logic (DL) [25].

OWL-S Ontology Web Language Services: an ontology built on top of Web Ontology Language (OWL) by the DARPA DAML program. It replaces the former DAML-S ontology. "OWL-S is an ontology within the OWL-based framework of the Semantic Web for describing Semantic Web Services. It will enable users and software agents to automatically discover invoke compose and monitor Web resources offering services under specified constraints".

PAA Piecewise Aggregate Approximation: a very simple dimensionality reduction method for time series mining. It minimizes dimensionality by the mean values of equal sized frames [88]. It is a method used to summarize sequences of time series data [119].

RDF The Resource Description Framework consists of a number of tools that use concepts from graph theory to add relationships and semantics to unstructured data such as the World Wide Web. The central aim for the RDF framework is to provide a way for machine interoperation of cross-domain data and merging information from different sources as effortless as possible. An RDF *triple* or *statement* is the foundation of the RDF data model. It consists of a subject a predicate and an object resource that together form a statement. Triples consisting of matching subjects and objects can be linked together to form an *RDF graph.*

RDF store or triple store a software system built on top of either a general purpose relational DBMS (Database Management System) or a custom DBMS which is mapped to handle the RDF data model. In addition to providing storage and retrieval of RDF data RDF stores can include a number of tools related to the Semantic Web such as reasoners graph exploration etc.

Reasoner A semantic reasoner reasoning engine rules engine is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language and often a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining [6].

Redland RDF store[7].

RuleML Rule Markup Language: a markup language developed to express

[6] `http://en.wikipedia.org/wiki/Semantic_reasoner`

[7] Redland Smart M3 v0.3.1-alpha: `http://sourceforge.net/projects/smart-m3`

both forward (bottom-up) and backward (top-down) rules in XML for deduction rewriting and further inferential-transformational tasks.

SIB Semantic Information Broker within Smart-M3 platform. The SIB is the central repository of information is responsible for information storage sharing and management through the Smart Space Access Protocol (SSAP).

Smart-M3 M3: Multi-domain multi-device and multi-vendor (M3) *Smart Space* platform that consists of independent agents which communicate implicitly by inserting and querying information in the space. *M3* is an open source cross-domain architecture composed by a SIB and KP entities that implement functionality and interact with the Smart Space by inserting/querying common information through the publish/subscribe SSAP protocol and also through SPARQL. *M3* supports RDF triple pattern queries as well as WQL and SPARQL queries. *Smart-M3* was originally developed by Nokia Research Center and a set of toolbox and applications were built around it among others at our Embedded System Lab. at Åbo Akademi within the DIEM project. *Smart-M3* continued being developed later on as an Open Source project by the open community called *Open-M3* and also by the Finnish-Russian University Cooperation in Telecommunications (FRUCT)[8]. In the thesis we refer to *Smart-M3* and *M3* indifferently.

SPARQL (pronounced "sparkle" an acronym for Simple Protocol and RDF Query Language) the W3C recommendation query language for querying RDF datasets. The SPARQL protocol is part of the query engine in most modern RDF stores and allows for a query to consist of triple patterns conjunctions disjunctions and optional patterns.

SQL Structured Query Language: a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).

SS Smart Space: A Smart Space is a representation abstraction of a ubiquitous physical and virtual environment in which heterogeneous devices share information to a common knowledge base as well as interact with each other. Some Smart Space implementations (such as Smart-M3) make use of semantic technologies.

SSAP Smart Space Access Protocol: a protocol implemented in Smart-M3 platform which provides the KPs access to the Smart-M3 space by means of the operations: Join/Leave the Smart-M3 space Insert/Remove information from the SIB Update Query and Subscribe to changes (triple patterns). It also allows to join and leave a concrete Smart Space [103, 192, 102]. Therefore the role of a module such as *Smart-M3* is to serve as SPARQL persistent storage for (crisp) event.

[8] FRUCT: `http://fruct.org/`

SW Semantic Web: "The Semantic Web is an extension of the current web in which information is given well-defined meaning better enabling computers and people to work in cooperation" [29]. SW is a collaboration of the W3C and others to provide a standard for defining data on the Web. The Semantic Web uses XML tags that conform to Resource Description Framework and Web Ontology Language formats (see RDF and OWL).

SWRL Semantic Web Rule Language: a proposed language for the Semantic Web that can be used to express rules as well as logic combining the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language (itself a subset of Datalog). SWRL includes a high-level abstract syntax for Horn-like rules in both the OWL DL and OWL Lite sublanguages of OWL. A model-theoretic semantics is given to provide the formal meaning for OWL ontologies including rules written in this abstract syntax. An XML syntax based on RuleML and the OWL XML Presentation Syntax as well as an RDF concrete syntax based on the OWL RDF/XML exchange syntax are also given.

UbiComp is an emerging paradigm for interaction between people and computers. A guiding principle of ubicomp is to break away from desktop computing to provide computational services to a user when and where required (using any device in any location and in any format). Another term for ubicomp is "Everyware" where computing is made to appear everywhere and anywhere.

UML Unified Modelling Language a general-purpose modeling language in the field of software engineering which is designed to provide a standard way to visualize the design of a system.

W3C The **W**orld **W**ide **W**eb **C**onsortium consists of member organizations staff and public participants that work together in an effort to create unified protocols and guidelines that will lead the Web to reach its full potential. W3C is the main international standards organization for the WWW; it was founded by Tim Berners-Lee in October 1994.

WQL Wilbur Query Language[9]: In WILBURQL queries are expressed as path patterns that match paths between a root node and the members of a result set. The path patterns effectively are regular expressions over properties of the underlying RDF data (i.e. they are regular expressions over the edge labels of the graph) [132].

WWW World Wide Web: an information system on the Internet which allows documents to be connected to other documents by hypertext

[9] Wilbur Query Language: `http://wilbur-rdf.sourceforge.net/`

links enabling the user to search for information by moving from one document to another.

XML Extensible Markup Language: a markup language much like HTML designed to carry data not to display data. XML is a W3C Recommendation designed to be self-descriptive. XML tags are not predefined and one must define his own tags.

Contents

Chapter 1

Background: Ubiquitous Computing, Smart Spaces and Ambient Intelligence

> Choose a job you love, and you will never have to work a day in your life
>
> Confucius

The notion of ubiquitous or pervasive computing, Smart Spaces and Ambient Intelligence are often used as synonyms within the area of Artificial Intelligence. However, some subtle emphasis can be made to characterize each paradigm, and better introduce the background to this thesis.

1.1 Ubiquitous and Pervasive Computing

The idea of ubiquitous space was proposed as an ideal world where humans and surrounding devices interact effortlessly. People would be surrounded by intelligent intuitive interfaces embedded in all kinds of objects and the environment would be capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way [11]. Thus, a transparent technology in the environment facilitates humans an easier everyday life [174, 76]. Mark Weiser [217] is the father of Ubiquitous or Pervasive Computing, and describes it as the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user.

1.1.1 Smart Spaces

In this thesis we will refer to Smart Spaces when we refer to ubiquitous computing environments and more specifically, when we want not only to observe and understand what happens, but also, when we have to react, intervene or act in the environment once the user's behaviour is understood.

A *Smart Space* (SS) [189] is any physical environment equipped with sensors and actuators able to perceive human activity and environmental conditions, to make decisions from these perceptions, and to modify the space according to the system goal. Smart Spaces support the vision in which computers work on behalf of users, they have more autonomy, and they are able to handle unanticipated situations. Therefore, the development of a SS implies the usage of Artificial Intelligence (AI) and machine learning, among other technologies.

Figure 1.1 shows an example of one implementation of Smart Space. Devices share information through a Semantic Information Broker (SIB), which is concrete to a specific implementation of Smart Space, called Smart-M3 [102]. This and other infrastructures and architectures for Smart Spaces are further described in Section 2.1.1.

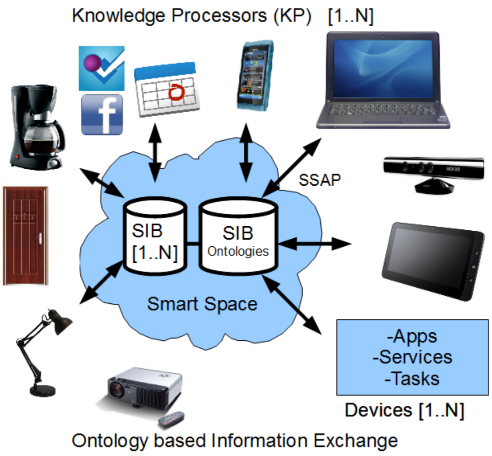


Figure 1.1: Smart Space and example of implementation (M3 [102]) and devices [189, 192]

Figure 1.2 shows the concept of a Smart Space with heterogeneous devices, and places the activity recognition cycle into context, considering the different phases it is composed of: event registration and classification, behaviour extraction, behaviour rules extraction, rule inference, behaviour recognition and user assistance or supervision. Activity recognition is the focus from Chapter 3 on.

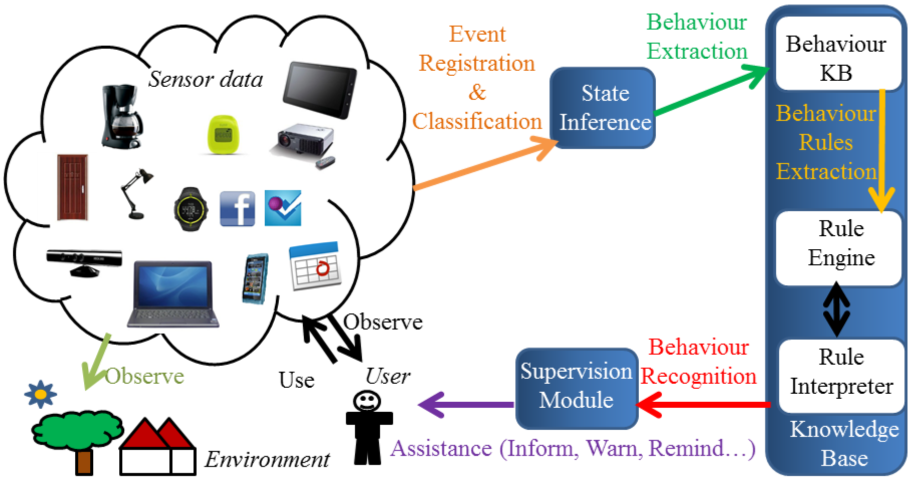


Figure 1.2: Ambient Intelligence. Smart Space and activity recognition cycle

1.2 Ambient Intelligence

The term **Ambient Intelligence (AmI)** was introduced by the European Commission in 2001 [110] as a response to new user needs in ubiquitous environments. One of the latest definitions [22] describes AmI as *digital and proactive environments, with capacity to sense the environment and assist users in their daily lives*. The major difference between *Ubiquitous Computing* and AmI is the introduction of Artificial Intelligence in the latter. Thanks to this, user-friendliness, user-empowerment, human assistance, and easy interaction with efficient services are improved. Examples of AmI scenarios can be seen in [22, 175, 55].

AmI systems are usually composed of at least: a) a *perception* mechanism to gather information from both the user and the environment [175], b) a set of *actuators* to modify the environment and communicate with users, and c) a *reasoning/decision making* module able to recognize what is happening to users in the environment, what they do and what their aims are, and to make decisions to assist them. These three abstract components often expand when designing AmI applications, since some scenarios also require the design of a sensor network, data fusion techniques, or a real-time response, among others [175, 169].

In AmI systems in general, the user occupies a central part. Thus, it becomes necessary to develop techniques to model, learn, recognize, and predict *what users are doing* in the environment, so that the system is able to make decisions about how to assist them. Usually, the literature calls *what users are doing human behaviour* or *human activity* interchangeably [175, 55, 171, 183]. These terms usually mean *a sequence of human actions* *that can be tagged with a label*, i.e. the corresponding activity/behaviour. However, most of these authors agree to define *human action* as the simplest unit in human activity, and it is usually associated with a sensor event. From our point of view, as long as new semantic approaches are being developed [48], new abstraction levels will appear in the system. For this reason, in our opinion, a difference will be made between the terms *human activity* and *human behaviour* to separate the concepts of what the user is really doing in the environment (activity), which is inferred from sensor data and machine learning techniques, and the purpose or meaning it could have (behaviour). However, we will not consider any difference between *human activity* and *behaviour* in the rest of the manuscript to preserve consistency with the existing literature.

Besides perception, actuation and AI techniques, other important part included in the design of AmI systems is the analysis of the most suitable task model required to achieve a well designed approach. In Activity Theory [155], the fundamental unit of analysis is human activity. Activities' aim is to accomplish a goal. Sampling, analysing, and modelling are examples of a user modelling process [42]. As the level of decomposition in task modelling depends on its purpose, task models should be rich in information and flexible to capture all the main activities that should be performed to reach the desired goals, as well as the different ways to accomplish them [10]. A summary of existing task models and their limitations can be found in [81]. *HTA (Hierarchical Task Analysis), GTA (Groupware Task Analysis), CTT (Concur Task Trees), UAN (User Action Notation), TKS (Task Knowledge Structure), DIANE+*, and *TOOD (Task Object-Oriented Description)* are examples of hierarchical, tree-based, and taxonomic structures. Common requirements often included in task models consider tasks, goals, activities, devices, time operators, etc.

Despite the great advances produced in the last decade, the complexity and the quantity of possible complex activities [153], the temporal interdependences among actions [183], the relevance of the semantics associated with a behaviour [48], or the existence and interaction of several actors in the same environment [196, 55], among others, make learning and recognition of human behaviour non trivial and raise clear challenges in AmI research.

1.2.1 Ambient Assisted Living

One concrete area within **Ambient Intelligence (AmI)** is **Ambient Assisted Living (AAL)**, whose intrinsic aim is oriented to user assistance. Human activity analysis is key here, where modelling of **Activities of Daily Living (ADL)** and complex behaviours, as well as assisting the user are the main objective.

Prominent projects in this area are *TigerPlace* [170], *iDorm* [74, 54], *CASAS* [171, 196] or [162]. They aim at combining the elderly independence with their constant supervision and required assistance. The technologies employed to acquire information depend strongly on the objective and activity to monitor. *iDorm* and *CASAS* employ embedded sensors and consider the hypothesis that, it is not necessary to identify the actor of each of the actions in real time, but learn the track of the behaviours through the detection of temporal moments and the areas where the actions happen [175, 196].

On the other hand, other research lines use wearable sensors, mobile devices, etc. [107, 228, 87]. [21] offers a broad study on sensor technologies on both approaches, embedded and wearable sensors.

Finally, another path recently open is the use of non intrusive video sensors, such as Microsoft's device Kinect, for indoor identification and tracking of multiple persons. Promising results have been shown when following several persons at the same time [53], although there exist limitations when the number of users rises or the distance of the user to the sensor is superior to about 3 metres.

As new approaches for task modelling include semantics to represent the meaning of human activity, context-awareness techniques become a more central part of AmI systems. To make this review more complete, the next subsection provides an introduction to the most widely used frameworks in context modelling.

1.2.2 Context-awareness: infrastructures and architectures

Context consists of any information that can be used to characterize the state of an entity [70]. Entities can include a person, an object, an environment, an application, or a device that interacts with the user. Context-awareness is one of the drivers of the ubiquitous computing paradigm and a well-designed model is a key accessor to the context in any context-aware system [204]. Proposals to model context can be integrated with human activity models provided with semantics. This subsection details models and frameworks to deal with context information as well as some ontologies used in these. The ontologies will be further studied in Section 3.2.

With regards to data integration [220], we may distinguish among widgets, networked services, and blackboard models. Widgets may hide low-level details of sensing and ease application development. On the other hand, networked services can be less efficient than widgets, but at the same time, they may form a more robust and flexible approach, e.g., by using a widget manager discovery in a context server architecture. Finally, blackboard models have a data-centric view with event subscription capabilities which provides simplicity for the addition of new data sources. A drawback is that the latter can have low efficiency in communication.

Concerning the way data is captured, sensors can be classified as physical sensors (i.e., hardware), virtual sensors (context data from software applications or services), or logical sensors (combining physical and virtual sensors). Different physical sensors are available for diverse types of contexts, e.g., light, visual context, audio, motion, acceleration, location, touch, temperature, or other physical attributes such as biosensors [26] or non-visual tracking systems in general [232].

There are different approaches [204] to conceptually model context. Key-value models [26] are one of the simplest approaches that serve to describe service capabilities in service discovery matching. Markup Scheme Models use hierarchical data structures with tags to define, for example, profile instances. Graphical models tools [204] such as UML (Unified Modelling Language) or extensions to ORM (Object-Role Modelling) are useful to appropriately and easily model context by means of graphical interfaces. Object-oriented models take advantage of OO features to encapsulate context processing and representation through well-defined interfaces. Logic based models are formal ways to represent facts, expressions and rules which allow an inference process to derive new facts based on existing rules. Finally, ontology based models describe concepts and relationships in a high and formal expressiveness level [26].

Ontology-based context modelling overcomes the limitations of other models regarding simplicity, flexibility, extensibility, generality, expressiveness, and automatic code generation [189]. Interoperability solutions based on the ontology model can benefit from ontology reasoning, since ontologies are the most promising and expressive models fulfilling requirements for modelling context information [26].

The survey [26] shows advanced context models in a good compendium of design architectures with their respective advantages and disadvantages. Georgia Tech aware house [121] is an example of a project that develops applications to support seniors living independently at home by using time independent heterogeneous context sources. Regarding context-aware frameworks, examples of OWL-based approaches for context modelling are *CoBrA* and *SOCAM*. *CoBrA* [44] is an agent-based infrastructure for context modelling, reasoning, and knowledge sharing using context acquisition components. *SOUPA* and *CoBrA-Ont* Ontologies are some of the related tools. User privacy control is also included. *SOCAM* (Service Oriented Context Aware Middleware)[84] introduces a server based architecture for building context-aware services focused on information sensing and context providers.

Another project providing an OWL encoded context ontology (*CONON*) is [222]. As an example, a reduced part of *CONON*, for the home domain, can be seen in Figure 1.3. *CoDAMos* (Context-Driven Adaptation of Mobile Services) [166] is another ontology, and it contains four main concepts: User, Environment, Platform, and Service. An excerpt of the CoDAMos ontology, with different environment conditions and locations, can be seen in Figure 1.4.

In *Gaia* [182], a metaoperative system is extended to include context-awareness. Instead of using RDF triples, *Gaia* uses 4-ary predicates (the 4th one is context-type), first order logic, and DAML+OIL. Other architectures for smartphone context-aware frameworks can be found in [133, 229, 213].

Context Toolkit [70] presents an approach to enable application development through reusable components. Situations are modelled on a system level, but there is no language level situation modelling. However, a restriction in meaningfulness exists due to its attribute-value tuples, in contrast to RDF. Many systems use SQL rather than the semantic standard SPARQL. Another example is *HIPPIE* [159], which utilizes existing users' information to distribute context information to their devices. *NESSIE* [168] focuses, on the other hand, on event based awareness. For compensating its lack of handling interaction, *HIPPIE* was combined with NESSIE [168], but the result still lacked semantic information description. In [30, 15], the Context Aggregation and REasoning (*CARE*) middleware interacts with the *COSAR* [178] system to recognize human activities through hybrid ontological/statistical reasoners executed on personal mobile devices.

In summary, we can observe different frameworks to facilitate the creation of context-aware services. They use distinct context representation models, different sensors and infrastructure. In addition, we observed critical research issues such as the type of context modelling and reasoning, knowledge sharing, and user privacy [44]. We may notice that there is a predominance for OWL languages. As we are interested in adding semantics to enhance the context-awareness in Smart Space to make the environment more intelligent, it is crucial to give non developers the power to program and control the environment so it reacts exactly in the needed moments. The next subsection studies context-aware configuration tools focusing on the end-user. *End-user* in this thesis is understood as any person with no technical nor programming background.

1.2.3 End-user programming frameworks for Ambient Assisted Living

Once context is captured, in order to provide the end-user the possibility of rapidly prototyping the behaviour of a Smart Space, abstracting away technical details but giving full potential and degrees of freedom, there are some challenges to face. When presenting a SS to the user with our requirements on having ontological capabilities, as it will be motivated in Chapter 3, the interface model, even if simplified, must adhere to the semantic formal model. Domain specific languages demonstrate support in this abstraction for integrating metamodels using ontologies (e.g. in [215]).

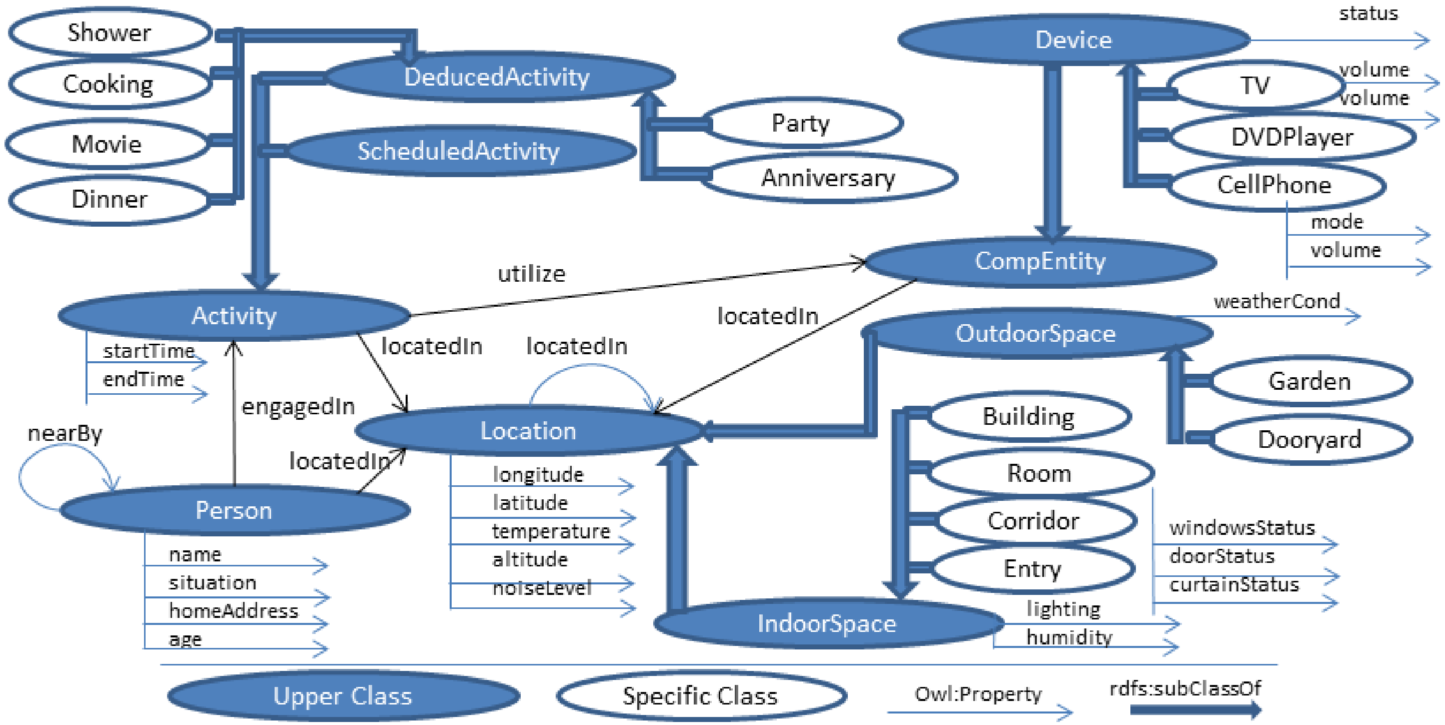


Figure 1.3: OWL encoded context ontology (CONON) [222] (partial)

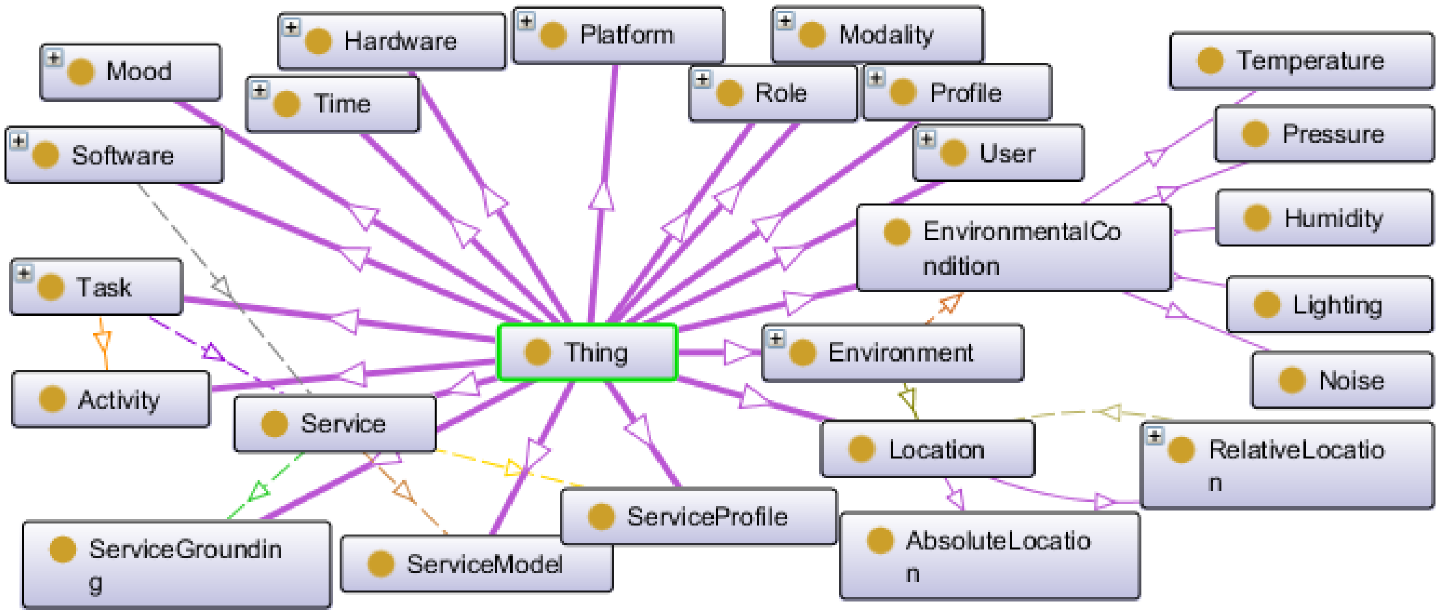
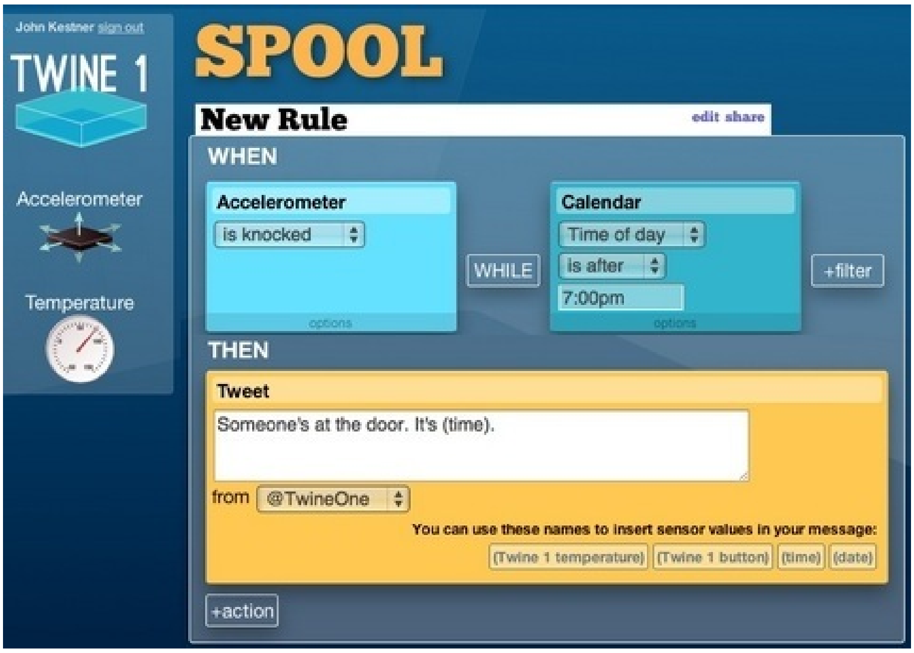


Figure 1.4: Ontology concepts within the CoDAMos ontology [166]

Concerning end-user GUI and rule editors, some good examples that simplify the tasks to the user when creating their own services/applications, through simple rules, are *If This Then That*[1], in Figure 1.7, for online social services, *Twine*[2] for sensor interaction applications, in Figure 1.5, or Valpas [177] intelligent environment for assisted living, in Figure 1.8.

Regarding the (developer/architect) user experience, another approach to aid the user interacting with semantic data consists of suggesting rules to the user by using browsed RDF paths, for rewriting these as WQL queries [132]. This is one of the new paradigms to be explored with SPARQL 1.1, for which the transitive closure allows concepts such as repetition, recursion and path queries.



Accelerometer: is knocked

Calendar: Time of day is after 7:00pm

Tweet: Someone's at the door. It's (time).

from @TwineOne

You can use these names to insert sensor values in your message: (Twine 1 temperature) (Twine 1 button) (time) (date)

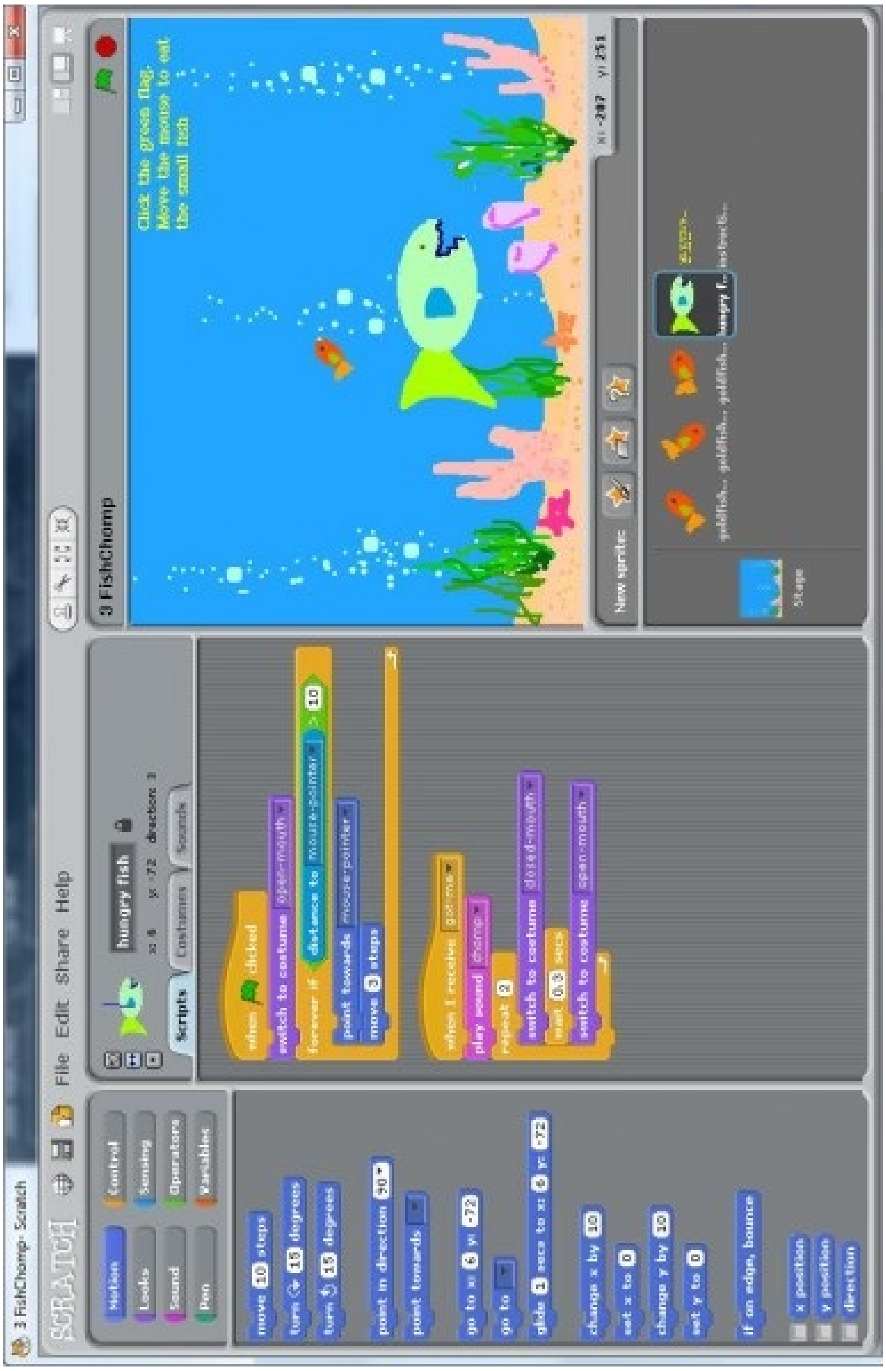
Figure 1.5: Twine Spool rule editor (http://supermechanical.com/twine/).

The survey of programming environments for novice programmers [117] shows ways to lower the barriers to programming. Also, *Scratch* is a recent successful programming framework [176] for kids (in Figure 1.6) that can serve as inspiration for easy and rapid application development.

Although the described frameworks have made more approachable the vision of the Internet of Things (IoT) and AmI for end-users, all these systems lack semantic capabilities to understand context though automatic ontological inference, as well as the possibility to deal with incomplete, uncertain

[1] `http://ifttt.com`

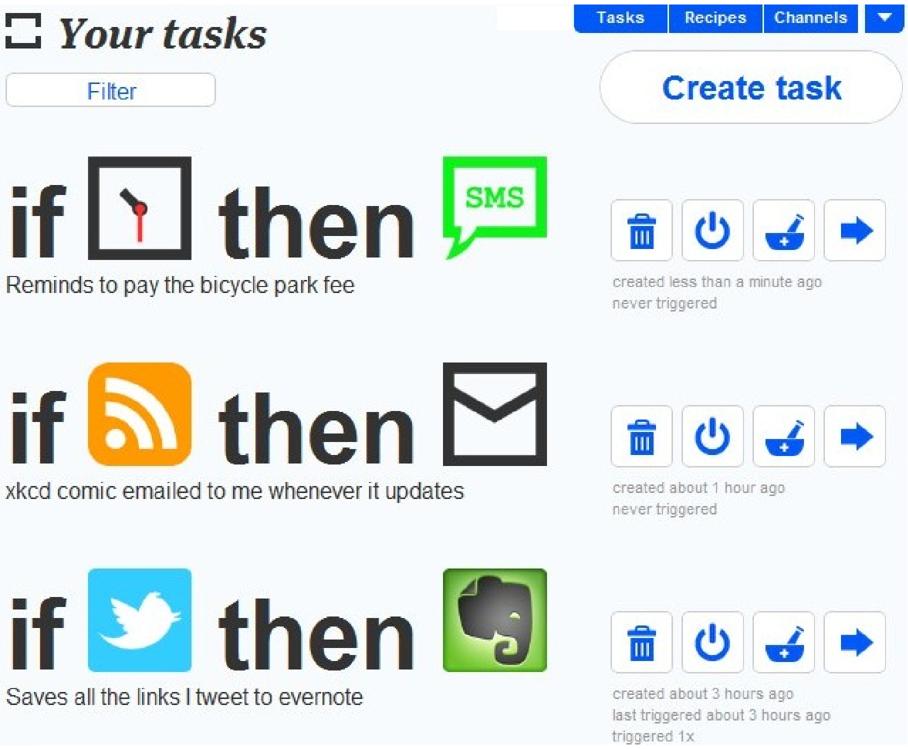
[2] `http://supermechanical.com/twine/`



```
when [flag] clicked
switch to costume open-mouth
forever if distance to mouse-pointer > 10
    point towards mouse-pointer
    move 3 steps

when I receive got-me
play sound chomp
repeat 2
    switch to costume closed-mouth
    wait 0.3 secs
    switch to costume open-mouth
```

Figure 1.6: Scratch [176] programming interface.



Reminds to pay the bicycle park fee

created less than a minute ago
never triggered

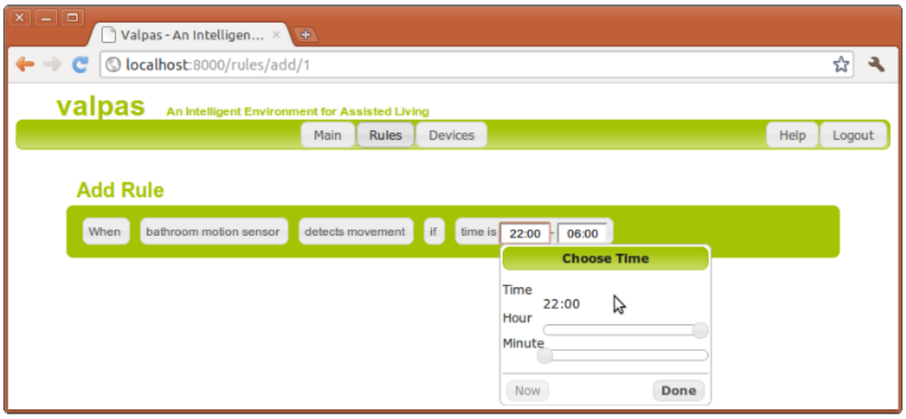
xkcd comic emailed to me whenever it updates

created about 1 hour ago
never triggered

Saves all the links I tweet to evernote

created about 3 hours ago
last triggered about 3 hours ago
triggered 1x

Figure 1.7: Example of IFTTT rule recipes (http://ifttt.com)


Valpas - An Intelligen...
localhost:8000/rules/add/1
valpas
An Intelligent Environment for Assisted Living
Main
Rules
Devices
Help
Logout
Add Rule
Choose Time
Now
Done

When | bathroom motion sensor | detects movement | if | time is 22:00 - 06:00

Time 22:00
Hour
Minute

Figure 1.8: Example of ECA rule editing in *Valpas*, an intelligent environment for Assisted Living [177].

or vaguely expressed information. Chapter 2 will tackle these features by proposing a new architecture that fulfils all these requirements. Specially, in Section 2.3, we propose a visual model taking inspiration from *Scratch* to aid customizing the behaviour of SSs with the added value of supporting semantic context-awareness and near natural language knowledge representation.

1.2.4 Objectives of the thesis

The work done in this thesis started motivated by the participation of Åbo Akademi University in the Finnish DIEM (Devices and Interoperability Ecosystems) TIVIT-SHOK Project from the Finnish Funding Agency for Innovation (2010-12). The objective was to ease the programmability of Smart Spaces with a multi-device, multi-platform, and multi-part open-source platform developed by Nokia Research Center (a project partner), to facilitate ubiquitous environments to behave proactively with users [189, 192]. Åbo Akademi's project responsibility, more concretely, was to achieve seamless device interoperability in Smart Spaces by providing tool-support to facilitate, with domain specific languages (DSL), the programmability and smoothness of ubiquitous computing applications.

To achieve our aim using the main backbone platform provided by Nokia, a general purpose rule-based domain specific language was developed in the Embedded Systems Laboratory in Turku, based on a tool set for SS application development. When further getting deep into the SS context, it was found that real world, imprecise and uncertain information needed to be taken into account to be properly handled. Fuzzy logic [230] was proposed as an ideal tool to handle possibility and uncertainty. This was the rationale to later propose a series of works on making the treatment of uncertainty accessible and automatic in AmI.

We found that few approaches focused on allowing the end-user to take part in deciding how the Smart Space should react to the user activities, i.e., when to provide help, notifications or alarms to the occupant of the Smart Space. In order to make the SS accessible and configurable not only by developers, but also by end-users, there are needs to provide a comprehensive framework that allows to program the behaviour of the SS to the developer, but also a visual interface for non-expert users to be able to configure the SS as well.

As the project developed, we aim at filling the gaps of existing solutions by providing the framework with the ability to withstand near natural language expressions, typical of everyday activities, accounting for uncertainty, incomplete or vaguely expressed data. As there are no frameworks dealing with each of these aspects regarding AAL in Smart Spaces, our first research questions are:

1. *Can we provide a programming toolbox for developers who are unfamiliar with Semantic Web and fuzzy logic to be able to benefit from these paradigms, by including it on regular programming patterns?*

   The technical semantic infrastructure to add meaning to the way *ubiComp* interacts in our lives already exist. However, we can facilitate the task of semantic programming with tools such as DSLs, code generation or uncertainty reasoning, by using W3C semantic and interoperable standards, together with development frameworks that ease this task.

2. *Can we also provide non-expert end-users (with no Semantic Web, fuzzy logic, nor technical background in general), the ability to control and program the Smart Space they live in, as they wish, and adapt it to their needs, so it reacts assisting in their daily lives?*
   An end-user visual language framework can allow non-expert users flexibility on expressing uncertain or imprecise data with near natural language. The result would make accessible the potential of context-aware and inference tools to empower the end-user.

   After analysing existing end-user tools and identifying missing features, Chapter 2 will tackle these questions by providing solutions from different points of view, focusing on the object-oriented developer through an OWL-based wrapper, and on the non-technical end-user through a novel visual language.

   Once developed the architectural frameworks to deal with context-modelling, gathering of heterogeneous sensor input data, and a publish/subscribe architecture with an interface for application development, we found that SW standards such as SPARQL, RDF and OWL reasoners do not intrinsically handle uncertain or vague knowledge, typical of ubiquitous environments and human behaviour. There is a need therefore, to accommodate the ability to reason about uncertain information, as this is a feature missing in regular RDF-stores. An exhaustive analysis on fuzzy OWL reasoners and how they can adapt to existing SW technologies (such as our pub/sub M3 architecture), is required to choose the best compromise on expressivity and versatility, while taking into account our AAL requirements. Therefore:

3. *Can we integrate the flexibility of fuzzy reasoners to the robustness and versatility of common crisp RDF storage infrastructures and include practical pub/sub SSAP-like protocols?*
   In order to answer this question, we proposed a mapping from traditional crisp architecture query languages (SPARQL) into the fuzzy paradigm [66] [65].

In any of the three programming cases (in previous 3 questions), it will be demonstrated that the development of Smart Space applications is not suitable specifically only to AmI environments (such as in [189, 192]), but also in other domains where interoperability of heterogeneous devices and context-awareness are key elements. As a sample, work was done on applications of the semantic interoperability framework on biomedical imaging [72], in a distributed scalable and low-power cluster [28], in remote rehabilitation within the Active Healthy Ageing platform to support health and well-being [63, 67] (in Section 4.2), and other scenarios [62], such as privacy and security [105] or interactive TV applications [191, 190] for, e.g., TV program recommendation. Some of these works will be commented as special case studies in Chapter 4.

Once we set the grounds to be able to develop SS applications to interact with the user in a efficient, interoperable and flexible way, the next step is interpreting sensor data in order to understand the meaning of what is happening in the SS, to be able to assist the user specifically, when he or the situation really requires so. Because human activity recognition is one of the most challenging but crucial tasks in AAL, it was the next target to properly understand what goes on in the space. A state-of-the-art in AR methods is required to undertake this problem in the most realistic way.

As it will be seen in Chapter 3, semantic strategies including ontologies are shown to be promising approaches in initial experiments on modelling and recognizing human activity. However, we find deficiencies among the existing ontologies. These limitations serve us to formulate research questions based on the motivational background section presented, i.e., after creating an appropriate context-aware approach in AmI for flexible end-user application development, suitable human behaviour models need to be created that are able to take full advantage of the proposed framework. Thus, the need for a generic full ontology that tackles missing domain entities in existing approaches for complex real-world environments, and that is able to deal with uncertainty in the same way as our infrastructure does, becomes evident. In other words, a research question would be:

4. *Can we effectively use semantic and ontology-based reasoning to recognize different level, simple and complex, real-life human activities?*

In order to assess existing AR domain ontologies, we studied available ontologies in a state-of-the-art, and found that the treatment of uncertainty, vagueness and imprecise information has not been handled inherently in the ontology representation [71]. Therefore:

5. *Can we provide the ability to handle automatically and in a natural way, imprecise, vague, uncertain or incomplete data (such as missing/wrong sensor readings) in real-time situations with ontological activity modelling and recognition? Can it provide more accurate results than classical crisp approaches?*

After choosing the most appropriate platform for SS application development, Smart-M3, we recognize the lack of support to work with fuzzy ontologies in most RDF architectures. Uncertainty representation in SS is therefore not possible to deal with in current environments. Thus, the next step required to achieve our objective was to design an ontology that can internally handle uncertainty in a natural and automatic way. This requirement made us switching from regular crisp RDF stores to fuzzy reasoners that inherently can deal with uncertainty reasoning.

The previous question lead to the proposal of a fuzzy ontology for human activity recognition in Chapter 4. Once a crisp and fuzzy ontology were proposed to provide missing sub-domains, and once we implemented several use cases with if/then rules on the office/work domain and in the exercise-workout domain, the next question is:

6. *Can the proposed approach be validated with a public, external and complex enough dataset of activities?*

When proposing a fuzzy ontology as methodology to AR, the questions to be assessed are related with the comparison with existing data-driven approaches in AR. Therefore, we need to validate if the benefits of knowledge-based approaches still hold in our problem, in contrast to data-driven methodologies. Thus:

7. *Is the method robust enough to simplify complexity in the training phase in case of new addition/removal/replacement of input data? Is it applicable to real life scenarios where the real-time ability to react is critical?*
   Chapter 7 evaluates the fuzzy ontology versus a crisp version of the same ontology for accuracy and execution time results. On a second experiment, a large input dataset shows real life implications on precision, accuracy and activity recognition time in more complex activity recognition.

8. *Is the semantic framework integrable with a traditional data-driven system for activity recognition, and able to improve the context-awareness interpretability, looseness and accuracy/precision of traditional methods?*

   In order to evaluate this question, experimentation and tests tackled not only the crisp features of the ontology, but also its main potential, i.e., the treatment of uncertainty regarding missing sensor readings, occlusions in image data or changes in the way the users perform the activities (different object usage), etc. (see Chapter 6 and Section 6.3). The experimentation chapter (Chapter 7) provides positive answers to these questions by providing a hybrid framework, where a data-driven DTW machine learning method developed in the research group is plugged on the bottom of the semantic high-level inference engine for assembling a whole hierarchical activity recognition module that takes advantage of the benefits of both data and knowledge-driven AR paradigms.

The thesis can be divided into two main segments or topics handled. Next chapter, Chapter 2, details the infrastructure and deployment side of Smart Spaces by presenting an architecture design and toolbox to provide a versatile approach to program Smart Spaces. This chapter handles three different points of view, considering the object-oriented paradigm to program the behaviour of the Smart Space, an architecture schema to improve interpretability, semantics and uncertainty treatment, and finally, a visual language for non-expert users to be able to control the behaviour of the SS. The remaining of the thesis handles the algorithmic and modelling part of the thesis and devotes to human activity recognition regarding the semantics and uncertainty aspects.

Therefore, the following chapter, Chapter 3, describes related work and a state-of-the-art on existing approaches employing data-driven and knowledge-based human activity recognition models. Advantages and inconveniences of these methods are discussed. In Chapter 4 we present a novel fuzzy ontology to deal with missing subdomains, identified in previous chapter, focused mainly on imprecision in human activity recognition. More concrete case studies include crisp ontologies to model human movement and interaction and privacy and security at low information levels. Chapter 5 presents how uncertainty, vagueness and imprecision are tackled with the proposed ontology for activity recognition and provides a case study on the office/work domain. Chapter 6 presents a two-fold hybrid activity recognition proposal; first the data-driven approach for (lower-level) sub-activities in Section 6.1, and then an ontology-based framework in Section 3.2. After detailing the hybrid architecture system, Chapter 7 documents two experiments for validation of both proposals, the fuzzy ontology against crisp approaches, and the hybrid framework as an overall hierarchical HAR system for tracking and recognition in real-time. Finally, Chapter 8 concludes with the thesis' contributions, while Chapter 9 discusses further remarks and future work.

Chapter 2

End-user rapid-development of Smart Space applications with near natural language reactive rules

> World needs fuzzy logic to reach end-users, otherwise it stays as our local hero
>
> Janusz Kacprzyk

This chapter focuses on the software practical implications and infrastructural mechanisms to achieve proactive AAL.

We propose a semantic model to represent human activity context and support imprecision, vagueness and uncertainty, as common characteristics of changing contexts, specifically when programming the behaviour of the SS. In order to facilitate the way to customize the behaviour of the Smart Space through simple rules we propose a visual language model that maps into two different development tools: a) a *PythonRules* middleware approach, for which learning OWL or SPARQL is not required to achieve device interoperability and interaction in the SS; and b) a fuzzy-crisp hybrid KB architecture to allow the exploitation of crisp or fuzzy KBs, according to necessities of the application.

In this chapter's first section, we first focus on framework *a*, then *b*, and finally we propose the visual language model. More concretely, the first section proposes a framework to allow end-users to program the behaviour of the Smart Space without requiring technical skills and allowing the flexibility of using natural language and imprecise information. Regarding the programming approaches, we first focus on abstracting away Semantic Web details to allow the programming of Smart Spaces using a regular object-oriented paradigm for software developers. As this still requires familiarity with programming and perhaps, keeping a graph-based knowledge model in mind, it was found that a big effort was still necessary to be done to allow non-expert users to configure the behaviour of the environment. Because of this, we took action in two ways. In the second section (Section 2.2), we provide an adapted architecture that better supports fuzzy natural language knowledge representation, while keeping the richness of ontological semantics. This approach uses fuzzy logic for more flexible, close to natural language interface.

Finally, in subsection 2.3, we concentrate on providing ordinary end-users with an accessible and functional SS vision through a visual language tool that uses the previous approaches to allow the exploitation of the potential of SW technologies, without requiring technical knowledge and supporting everyday life tasks, in order to program the behaviour of the SS.

2.1 Programming the behaviour of Smart Spaces with *Python* reactive rules and a publish/subscribe architecture

In pervasive and context-aware computing, a user should be able to readily accomplish an action which possibly can include cooperation and collaboration with others using multiple devices and networks as he moves in the environment. In this way, a whole new universe of intelligent applications would automatically adapt to the user's intention.

Let us assume a user's favourite program starts in 5 min. based on a profile information or a fan page on *Facebook* and the TV guide available on the broadcaster's web page. Then it could use GPS to find that the user is not at home and start the PVR (Personal Video Recorder) at home. This kind of intelligent applications need the context information from different sources to adapt to the user's preferences without involving human interactions. The context-aware intelligent applications can be realized by exposing the context information, internal data and functionality of the devices and ensuring data interoperability between them. This requirement is due to the variety of devices to be used and the need for interacting with each other within the context.

To enable the above mentioned cross-domain scenario and to solve the interoperability issue, one way is through the notion of *Smart Space*. A Smart Space is an abstraction of space that encapsulates both the information in a physical space and the access to this information allowing devices to join and leave the space. In this way the Smart Space becomes a dynamic environment whose identity changes over time when a set of entities interact with it to share information. For example, communication between the mobile phone and the PVR in the above scenario does not happen point-to-point but through the Smart Space whose members are the mobile phone and the PVR.

We have developed a programming interoperability solution for rapid application development in Smart Spaces that can be extended to support context-aware intelligent applications [116].

2.1.1 Smart-M3 Platform

Smart-M3 is a Multi part, Multi device and Multi vendor platform[1] that consists of independent agents which communicate implicitly by inserting and querying information in the space. *M3* is an open source, cross-domain architecture where the central repository of information, Semantic Information Broker (SIB), is responsible for information storage, sharing and management. Entities called Knowledge Processors (KPs) implement functionality and interact with the Smart Space by inserting/querying common information through the publish/subscribe Smart Space Access Protocol (SSAP) and also through SPARQL. *M3* supports RDF triple pattern queries as well as WQL[2] and SPARQL queries. Table 2.1 compares the different query operations for SPARQL and SSAP protocols. Although RDF is underlying both models, the main difference lays on extra functionality, in SSAP, for joining/leaving a confined SS data store, as well as the subscription advantage, for awareness of data changes. However, this capability easily became a performance bottleneck, but efficient implementations were recently developed[3]. *Smart-M3* was originally developed by Nokia Research Center and we built a set of toolbox applications around it. *Smart-M3* continued being developed later on as an Open Source project by the open community, and more commonly refer to as *M3*.

Figure 1.1 (in Chapter 1) showed a Smart Space architecture implemented using M3. Communication happens not device to device but through the SIB, while entities and services are described with OWL (Web Ontology Language).

M3 has some clear benefits. The main one is the subscription mechanism that avoids constant launching of queries to evaluate if certain condition satisfies. This feature is missing in the majority of semantic RDF stores. Benefits of publish/subscribe (or "push") semantic architectures, such as *M3*, include the inherent polling and a strong decoupling of the communication clients with respect to time, reference and data schema, increasing

[1] Smart-M3 platform, available for download: `http://sourceforge.net/projects/smart-m3/`

[2] Wilbur Query Language: `http://wilbur-rdf.sourceforge.net/`

[3] Redland Smart M3 v0.3.1-alpha: http://sourceforge.net/projects/smart-m3

Protocol comparison for SPARQL and SSAP query languages	
SPARQL Query	**SSAP Query**
-	*Join* -Join a named Smart Space
-	*Leave* -Leave a Smart Space
INSERT [DATA] [INTO]	*Insert* -Insert information to Smart Space
SELECT	*Query* -Query for information in Smart Space
DELETE [DATA][FROM]	*Remove* -Remove information from Smart Space
UPDATE	*Update* -Update information in Smart Space
-	*Subscribe* -Set up subscription (persistent query) to receive notifications when data changes
-	*Unsubscribe* -Cancel an existing subscription

Table 2.1: Protocol comparison for SPARQL and pub/sub-based SSAP query languages

flexibility in application design and allowing for more autonomous system architectures [152]. Other advantage of M3 is complementary support to semantic web standards, i.e., the original SSAP protocol which allows apart from insert, update and remove triples, subscribe and unsubscribe to a triple pattern, join and leave a concrete SS. Therefore, the role of a module such as *M3* is to serve as SPARQL persistent storage for (crisp) event processing.

2.1.2 A framework to develop Smart Space applications using Smart-M3 RDF store

We use an ontology-driven development approach with M3 to map ontology-based models to object oriented programming [Ont][116]. The approach consists of two parts. The first part, developed at the laboratory [116], is the generator that creates a static API from an OWL ontology. This mapping generates native Python classes, methods and variable declarations which can then be used by the KP developer to access the data in the SIB as structured and specified in the OWL ontology. The second part is the middleware layer which abstracts the communication with the SIB. Its functionality is the handling of RDF triples (*Subject, Predicate, Object*) with the generated API. This consists of inserting, removing and updating triples and committing changes to the Smart Space. It also provides functionality for synchronous and asynchronous queries. Our approximation enables application developers to use the generated API to develop new KPs and applications without worrying about the SIB interface as the generated API takes care of the connection to the SIB each time an object is created.

In this application development approach, the concept of application is not the traditional control-oriented application running on a single device, but rather a number of independently operated KPs which may run on different devices and are grouped together to be perceived as a single application. For instance, chat, calendar synchronization and multi-player games are examples of applications using this approach where a set of KPs, each handling a specific task, run on multiple smart devices and coordinate and interact with each other through the SIB to make a complete application. This coordination between KPs is done in the form of data exchange through the SIB where KPs subscribe to or query for specific data to perform a specified task. Application ontologies are used to describe data in the SIB and directs the KPs to access and manipulate data related to their functionality.

The developed programming interoperability solution for rapid application development in Smart Spaces is based on Nokia open source Smart-M3 architecture [102], as an ideal choice for developing pervasive applications as it includes: 1) A blackboard software architecture which is cross-domain, cross-platform and enables knowledge share and reuse. 2) Ontology governance process (information stored in RDF) ensuring seamless information interoperability.

In the following use-cases we created an ontology containing rules for automation, concepts for expressing the house state, and the temperature. It illustrates an application development approach for *Smart-M3* where the KPs are developed from the generated ontology API and are able to communicate through the *Smart-M3* space providing interoperability between different devices in the example application. The ontology components and their attributes were edited with *Protégé* [104], while the Python Code Generator was used to generate the agent ontology API with populated instance properties.

In order to address cross-domain scenarios such as the previous example at the beginning of this section (on a user's favourite program), technical and conceptual problems arise. The concept of *Smart Space* appears to encapsulate and abstract information from different services with the aim of allowing heterogeneous service composition.

A PVR could be considered as a form of API with different functions. One or several KPs can be perceived as a service, for example several KPs handling calendar activities in an application could shape a calendar service. Thus, each service provider exposes its functionality to other KPs and services through the *Smart Space*. At the same time each service acts as requester too. In this way, we could have the PVR's KP and the mobile phone's KP connected to the SIB. Figure 2.2 shows the registered devices' KPs with their information described in their respective ontologies.

Each of these subservices within a device inform of their inputs and out-

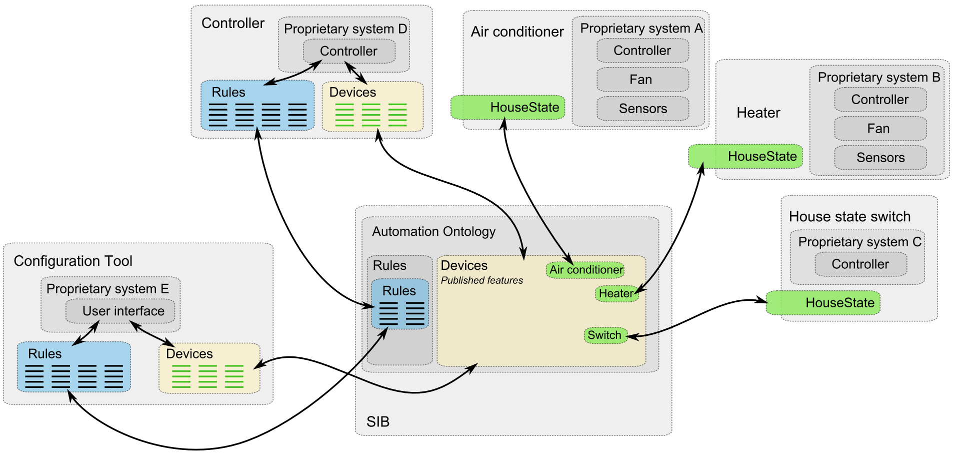


Figure 2.1: Case study overview example on home automation, showing an interoperability solution

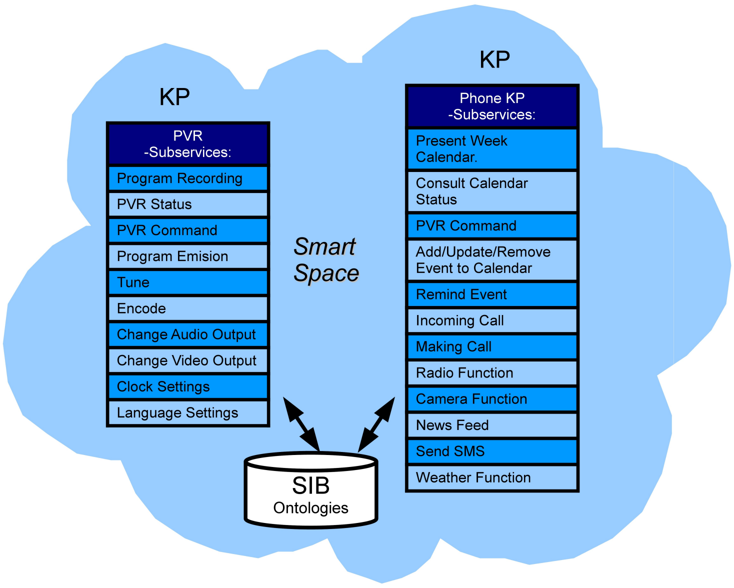


Figure 2.2: Knowledge processors (KP)' services structure

puts among other parameters in each of their profile. In order to deploy the scenario of recording the favourite program, the composition of required services must be deployed in the SIB, which knows about the devices connected to the *Smart Space*. But here it is found the problem that the SIB offers a persistent data repository but is a plain database giving just access to the data; no control structure or computation is provided. However, if we add to the SIB a description of each subservice, all devices would be represented with a unique standard allowing language and device independent service composition. For this purpose we can suggest service description with OWL-S [OWL] representation, because OWL-S enables declarative advertisement of service properties and capabilities that can be used for automatic service discovery and because it describes the services in terms of capabilities based on OWL (as well supported by *Smart-M3*). In addition to provide specification of prerequisites of individual services, OWL-S language describes services composition including data flow interactions [144]. With this purpose all atomic data sources should specify their functionality in terms of input and output data types as well as other meta information such as how long is the validity of its data, how accurate is the data, its nature i.e data is sensed or defined etc. This can be done by using Web Ontology Language for Services (OWL-S) [OWL] which has capability to specify characteristics and functionalities of all the information sources. OWL-S would facilitate Context DataType Interpreter to easily map data values from heterogeneous devices. More on our work done in this area is in [192].

2.1.3 Context ontology model and system architecture

Among the context modelling approaches existent, we choose ontology-based context modelling because of several reasons. Firstly, as M3 architecture provides an interoperability solution based on ontology models, we can benefit from automatic code generation and ontology reasoning. Secondly, ontologies are the most promising and expressive models [26] fulfilling requirements for modelling context information. Thirdly, ontology-based models provide advantages of flexibility, extendibility, genericity and expressiveness which are key factors in context-aware systems.

Information about the user's context is significant if enables the ambient system and applications to adapt to the user's preferences. We refer to all information that characterize the situation of a user as his *context* [189]. In order to make the system more adaptive to the user's behaviour, we propose to use multiple dimensions as the context of a user. Figure 2.3 shows these dimensions in an example context ontology. We divided the user's context in two broad categories, namely *atomic context* and *inferred context.* Atomic context refers to the context data acquired directly from context providers. The sources of atomic context can be any source providing rele-

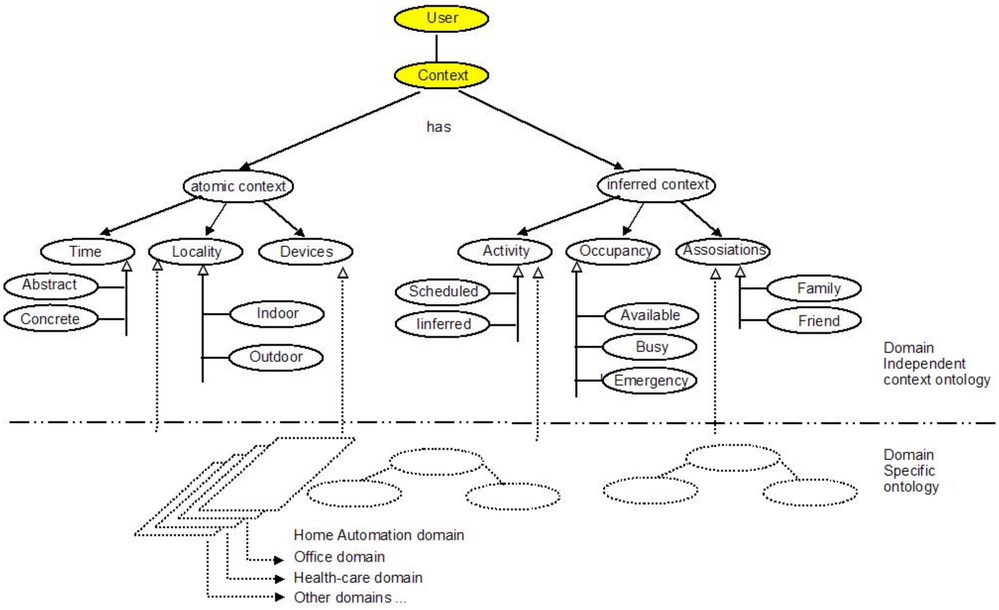


Figure 2.3: Context Ontology of a user

vant information to describe a user's situation. The inferred context refers to the information deduced from the given context data. We modelled user's context using six context dimensions: *Time*, *Locality*, *Devices*, *Activity*, *Occupancy* and *Associations*. Although this is not the only set, we believe that it is enough to capture most of the concepts in reactive applications.

As the context ontology defines the basic terms and relations related to the user context that are applicable in every possible domain, we have defined it as an independent layer as shown in Figure 2.3. The user interactions involve a number of devices and appliances available in the environment which make the context dimensions of the ontology consider their activities and associations as domain independent types. The upper ontology in Figure 2.3 represents the core concepts to model user's situations in the environment and it appears feasible to have an unified upper context ontology capable of dealing large communities of users in wide range of domains. The lower part of Figure 2.3 shows the domain specific ontologies which describe concepts related to the domain in question. For instance, the tasks which are performed under the office domain such as *meeting, presenting* etc. are different when compared with tasks in the home-automation domain.

The system can thus, through rules, deduce information not explicitly given in the ontology, e.g. if the user is in the living room and the TV is ON then it implies that the activity is watching TV. Similarly, the system can deduce that the user is busy if he is talking on the phone, even if his calendar shows no activity at that time. In this way, by using the context information from different dimensions, the system can adapt to the user's current behaviour and make the decisions rather dynamically.

The context reasoner or rule interpreter is responsible for inferring new higher level context from given atomic context information. The context reasoning is based in this case on inference rules defined by KP developers which are then provided to the Python Rules Module. It enables the context-aware system to be tailored for specific application scenarios. For example, if a user is in his bedroom, the bed sensor is On and the light is Off, the reasoner can infer that the user's activity is *sleeping* and put this inferred context information in the Semantic Information Broker. The context reasoner can also infer context properties using ontology reasoning by specifying inter-ontology relationships.

2.1.4 Inference rules and context reasoning

In context-aware ubiquitous systems where the emergence of increasing number of devices are used to perform desired services, we need to impose control constraining the participating devices' behaviour. Rules can define how to react when a phone rings during a meeting; how to handle multiple requests to play different channels on a single TV at the same time; how to infer the user's activity using the active context information from multiple KPs. The inference rules, based on logic programming, allow context information origination from the provided set of ontologies. Its evolution/adaptation is caused by KPs taking part in the application.

We can define an inference rule using a 3-clauses pattern. The *With clause* models declarations and assertions, the *When clause* contains events that trigger the rule, and the *Then clause* includes conclusions representing inferred information after the rule is triggered.

Following there are few examples which illustrate our approach to define inference rules. The first rule states that if the user is in room B4050 at a specific time 13:20, and the room is occupied between 13:00 to 15:00 having more than one person there, then the user's activity is inferred as *busy in a meeting*.

```
with U:- User(id="1", role="Student", name="Mohsin"),
     R:- Room(room No.="B4050", location= "ICT"),
     P:- projector (id="101", type="ProModel")
when U.locatedIn (R, atTime"13:20"),
     R.occupied("13:00-15:00"),  P.locatedIn:- R,
     R.number of people > 1, P.statusON
then U.busyInMeeting
```

The inferred context information can then be used by another rule to infer other level information or to perform a task when some event is triggered. For example, the following rule states that when the user is in the meeting then forward incoming calls to his voice mail without interrupting him.

```
with U:- User(id="1", role="Student", name="Mohsin"),
     Ph:- phone(id="10", type="Iphone", model="4G")
when Ph.incomingCall, U.busyInMeeting, Ph.owner:- U
then Ph.activeVoicemail
```

There might be some emergency cases when the user does not want to ignore incoming calls. The following rule overwrites the result of the previous one when the calling person is the user's wife giving a beep to the user's phone. The user's relationship with the caller can be obtained from existing ontologies given to the system, such as Friend of a Friend (Foaf) ontology in this case.

```
with U:- User(id="1", role="Student", name="Mohsin"),
     Ph:- phone(id="10", type="Iphone", model="4G"),
     C:- Caller(name="Samra", association="wife")
when Ph.incomingCall,  U.busyInMeeting,
     Ph.owner:- U,  U.relation:- C
then Ph.beepOnce
```

2.1.5 Development framework: programming knowledge processors in *Python*

Python's meta-programming features are used to enable writing Python code which includes logic programming statements representing inference rules. These are inspired from the event-condition-action (ECA) rules model which is a technology from active databases for supporting dynamic functionality [112].

The first task is defining a script language to use OWL 2 allowing the user to express rules as the previous subsection showed. The second task is the integration of those logic expressions to work with ontologies into a functional OO-language. Because of its versatility, meta-programming opportunities and ease of prototyping (easy to learn and use) we chose Python. Thus, given a context, the programmer could define in a simple way and beforehand the underlying rules that pervasively help the user daily in his Smart Space.

The third task is the integration of first and second approaches with the *Smart-M3 Ontology to Python API Generator* framework [Ont], which makes more intuitive to the programmer the definition of pervasive applications. This tool provides automatic generation of a Python API for every OWL class as well as setters and getters among other methods to interact effortlessly with the common SIB through which all KPs communicate with each other.

Given the functionality provided by the Smart-M3 Ontology-Python framework, there is a need for designing a rule syntax language that allows users -with basic programming skills- easy definition of rules to model pervasive applications. In this way the need for learning OWL or query languages is minimized or null. The main feature of the Python Rules Module is to encapsulate, acting like a bind, the SIB interface. Our implementation approach is inspired by *Pythologic, Prolog syntax in Python* [PyL]. A Rule is structured as follows:

```
With() |= When() >> Then()
```

- **With()** Clause represents *assumptions* about existence of individuals.
- **When()** Clause represents *conditions*, when the KP must execute.
- **Then()** Clause represents *actions* to trigger.

In this way, the application programmer does not deal with RDF triples directly but mainly with logic Python expressions. Therefore, the programmer could embed into Python code expressions like:

```
condition1 = lambda: user.isBusy()
condition2 = lambda: room.getOccupied()
conditions = [condition1, condition2]
action =  lambda: user.setVoiceMail(True)
myRule = With([user, room]) // When(conditions) >> Then
    (action)
diem.addRule(myRule)
```

Listing 2.1: Rule definition with Python Rules Module

The underlying implementation of the Python Rules Module translates Python logic expressions to the SIB API main interface: *Query, Subscribe, Insert, Remove, Update.* Thus, the Python Rules Module just needs to be imported to be used with the KP class where the SS application is coded:

- **With()**: If instances in *With()* exist in the SIB (SIB-Query), it proceeds to evaluate *When()*. The check includes the ontology's Python object declaration, i.e., other KPs know about it.
- **When()**: If *When()* is true (SIB-Query), executes *Then()*. If not, sets a SIB-Subscription to the attributes in *When* clause. The subscription capability provided by the Smart-M3 SIB allows knowing when the value of certain attribute has changed so that the rule can be evaluated again avoiding, in this way, unnecessary infinite query loops or traffic bottlenecks.

- **Then()**: If *With()* & *When()* satisfy, executes *Then()*, which translates into SIB-Update/ SIB-Add/ SIB-Remove/ SIB-Unsubscribe of RDF triples.

A Knowledge Processor can be located e.g. in any smart phone or device and can be for example a phone application for getting the local temperature from the Internet or a sauna/thermostat activator. All the KPs can be created and connected to the Smart Space (called ’x’ in this example) in the following way:

```
def main(args):
app = QtGui.QApplication(sys.argv)
smartSpace = (’x’, (TCPConnector, (’127.0.0.1’, 10010))
    )
KP = PhoneKP.create(smartSpace)
# Definition of Rules
sys.exit(app.exec_())
```

Listing 2.2: KP Programming and Connection to the Smart Space ’x’

Straight after the KP is created, the user could define Python rules related to the existing KPs.

If `EmptyKP.py` (provided by the Ontology-Python Generator [116]) is used, instance declarations will automatically translate to insertions of triples into the SIB. This allows other KP applications connected to the same Smart Space to know about those individuals’ existence to interact with them. In the Python Rules Module, every KP application contains a `TripleStore` instance (produced by Ontology-Python Generator) representing the Smart Space’ SIB. At last, the *With()*, *When()* and *Then()* Python clauses translate into one of the implementation options given by the Ontology-Python Generator (SIB calls in *RDF* or *WQL* language). Our approach shows that learning OWL or query languages is not necessary to connect to the SIB and with other devices’ KPs [189]. This section considered crisp-based only ontology-based reasoning. Next section will consider the uncertainty aspect when programming AmI systems.

2.2 Handling uncertainty reasoning when programming Smart Spaces

When defining and modelling human activity to ultimately achieve behaviour learning and recognition in Smart Spaces, its characterization is not intrinsically crisp. However, the data stream of events, i.e., the SS’ input that serves as base to model these human behaviours, is crisp. Therefore, we identify a gap, where connecting the qualitative view of SSs with the quantitative approach of SSs (i.e., the machinery around SS frameworks and architectures such as RDF stores and query languages) seems evident. Joining quantitative and qualitative paradigms would be crucial for expressing behaviour and adapt the SS to the user.

A paradigm gap appears among the quantitative aspects of SS, i.e., the infrastructure and technology that handles large amounts of crisp sensor information [146, 212, 153, 123] and the qualitative aspects of SS, i.e., the techniques for human activity modelling and recognition [115, 183, 184], and the formal specification of how the system behaviour should be according to what the system is perceiving [179, 13, 189]. The latter approach involves reasoning about imprecise and vague information. In this way, our purpose is to develop a new methodology able to entitle end-users to work with a vague semantic specification of the SS. This new abstraction level will allow to manage known human activities, their relationships and environmental conditions, to provide expressiveness to the specification of the SS behaviour, and therefore getting closer to the end-user's everyday language.

Regarding the quantitative view of SSs, we identify the need for a semantic store with support for standards (OWL 2, SPARQL, etc.), as well as a scalable enough rule engine with subscription capability for real time rule-based applications. By *scalability* we mean the capability of handling a wide variety of heterogeneous events and data from different users and activities. Supporting the standards is necessary to represent fuzzy ontologies, since nowadays, each reasoner uses its own fuzzy DL language [34]. We argue that the integration of these quantitative and qualitative requirements is important to precisely model human behaviour in SS, as well as to ease the development and deployment of semantic and intelligent applications. However, to the best of our knowledge, there is no integrated reasoning and storage solution supporting all our requirements.

In this section we will describe a hybrid architecture [66] that takes advantage of both paradigms. The first step is identifying the most suitable fuzzy reasoner in order to provide automatic and expressive uncertainty treatment. We therefore study available fuzzy reasoners and their flexibility next.

2.2.1 Fuzzy reasoners

Considering suitable infrastructure for Smart Spaces, we find a great offer in storage as well as reasoning solutions that push the Smart Spaces vision forward. Our requirement for these solutions is to be able to handle event subscription for data scalability. With regards to the fuzzy paradigm, imprecise knowledge reasoning, fuzzy Description Logics appear as an alternative to crisp DLs which lack the ability to represent uncertain or vague information. We consider a set of available fuzzy reasoners and evaluate a set of expressibility requirements, useful for AmI applications. Furthermore, we identify SPARQL support as a standardization requirement. Table 2.2 summarizes the identified requirements for behaviour representation in SSs and the fuzzy reasoners that support them to some extent (marked with x). It can be seen that *FiRE* and *DeLorean* allow the use of some DL constructs that *fuzzyDL* does not support (cardinality restrictions and, in the case of *DeLorean*, also nominals). On the other hand, *GURDL* supports a more general representation of uncertainty, not being limited to fuzzy logic [33]. At last, another useful feature in behaviour modelling is stream reasoning to semantically annotate events. This feature can be found in *TrOWL* [208], a tractable reasoning infrastructure of OWL 2 with built-in OWL 2 QL reasoner Quill and EL reasoner REL. *fuzzyDL* supports a series of distinct features with respect to expressivity of the representation, such as explicit fuzzy sets, concepts modifiers, data types and defuzzification [33]. However, none of the fuzzy reasoners includes a listener/observer subscription mechanism for effective changes notifications on real time.

Reasoner	**Fuzzy DL**	**Event Subscript.**	**SPARQL**	**Cardinality Restr.**	**Fuzzy Sets**	**Concept Modifier**	**Fuzzy Data Type**	**Defuzzification**	**Fuzzy Rule**	**Satisfiab. Degree**
FiRE [200, 199, 195]	$\mathcal{F} - \mathcal{SHIN}$			x						x
GURDL [91]	$\mathcal{F} - \mathcal{ALC}$									x
De-Lorean [32]	$\mathcal{F} - \mathcal{SROIQ}$			x	x	x	x			x
GERDS [92]	$\mathcal{F} - \mathcal{ALC}$									
fuzzyDL [33]	$\mathcal{F} - \mathcal{SHIF}(\mathbf{D})$				x	x	x	x	x	x
YADLR [127]	SLG algorithm									x
Fuzzy OWL Plugin[Fuz, 34]	$\mathcal{SROIQ}(\mathbf{D})$									
FRESG [94]	$\mathcal{F} - \mathcal{ALC}(\mathbf{G})$						x			x
SoftFacts	$\mathcal{F}$−DLR-lite									

Table 2.2: Comparison of available fuzzy reasoners and their support for Smart Space modelling requirements

No fuzzy reasoner includes at the moment subscription features; it is only in crisp RDF stores where this can be found. To the contrary, it is uncommon to have RDF stores including imprecise reasoning. However, there are particular instances such as *f-SPARQL* [52], a "flexible extension of SPARQL", that allows in the *FILTER* constraint, the occurrence of fuzzy terms and fuzzy operators (by using α-cut operation), as well as weights in fuzzy constraints to have different importance and efficiently compute the top-k answers. Other extension of SPARQL is *AnQL* (Annotated SPARQL) [234], which provides a general framework that enables querying annotated graphs. The annotation domains include temporality, fuzziness, trust, or multiple provenance queries.

Event subscription is not supported in the vast majority of RDF stores. However, we can find exceptions such as *M3* [102], some versions of *OWLIM* [owl] or *RDFStore-js* [98]. The latter is a JavaScript implementation of an RDF quad store with support for SPARQL 1.0, most of SPARQL 1.1/update and a significant portion of SPARQL 1.1 query, that can be executed in the browser. The great advantage of event subscription features in semantic repositories is the capability of efficiently get notified when data of interest changes. This feature can avoid bottlenecks normally caused when rule conditions result in a constant checking for the status of specific nodes.

To the best of our knowledge, and as Table 2.2 shows, there does not exist a system which comprises support for all our requirements: expressive fuzzy queries, event processing for scalable, efficient and real time applications, as well as the possibility of federating queries with other SPARQL end-points. These are crucial elements when tackling real life problems in SS. For instance, decision support system or expert systems must react on time against forgotten actions, against potential errors produced in an industrial process or when following up a certain procedure with guidelines. To solve these problems, not only must standard query languages be supported to integrate heterogeneous data, but also efficient notification mechanisms must be supported to be alerted only when conditions we are actually interested happen. Additionally, support for every-day imprecise or vague terminology is to be provided. Lacking an efficient subscription/notification based system makes a large rule system impractical to real-time proactive applications. The requirements detailed in this section, together with an easy and reachable end-user language, can allow better modelling of expert knowledge and better involvement of the end-user into the problem modelling process. In next section we discuss concrete components for our proposal and how they can be connected.

2.2.2 Crisp to fuzzy OWL query mapping to improve semantics and usability

After identifying, in the previous section, the main components required for realizing more powerful human activity modelling in SSs, we detailed some technologies which contribute to this aim. As no system was found fulfilling all the needs for the development and deployment of our vision of

SSs, we suggest a configuration of technologies that allows us to construct an AmI framework to enable human behaviour representation and recognition through rules that include fuzzy concepts in form of linguistic terms.

The overall system, in Figure 2.4, is composed of two parts, a crisp KB and a fuzzy KB. These are connected by a main rule engine, which handles the subscription to each type of event condition (fuzzy and crisp) per rule. Next sections detail the components of the system architecture.

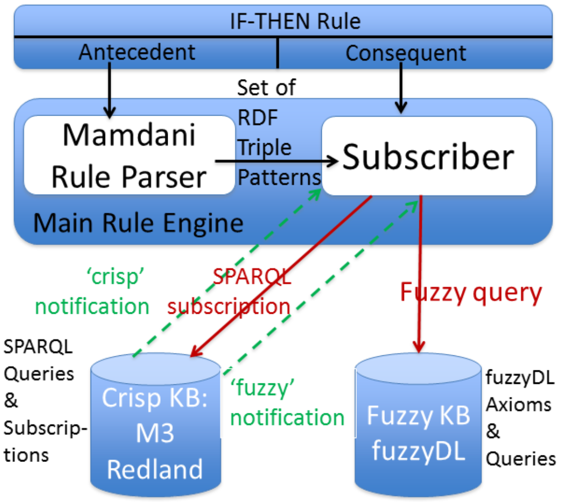


Figure 2.4: Overall framework with fuzzy and crisp Knowledge Bases

Once described, in last subsections, the main components of the reactive context-aware SS architecture, we proceed to study its integration within an event based hybrid rule-based system. Figure 2.4 shows the structure and main processing modules of the fuzzy-crisp overall architecture, as well as the information flow.

Crisp RDF infrastructures can achieve scalable SS applications. However, these are features not always considered to be the main aim of fuzzy reasoners. The latter, on the contrary, provide expressive languages to model, e.g., routine activities or more complex processes. In order to allow not only crisp but also fuzzy rules for describing the behaviour of both, users and a semantic SS as a whole, the SW needs to become more imprecise to accommodate everyday problems and serve distinct kinds of users [161]. This is why the user should be able to vaguely or imprecisely express knowledge. We proceed to explain how a rule with imprecise concepts and/or relations, is mapped to a representation in our fuzzy KB.

In order to reason with human behaviours, as well as the behaviour of the Smart Space system as a whole, we can employ an expressive (allowing necessary DL constructs as well as flexible behaviour descriptions) fuzzy DL reasoner, such as *fuzzyDL* [33]. Let's assume the user wants to define rules such as:

1. IF (*WeatherSituation isCurrently VeryStormy*) OR (*Natalia hasStatus AwayForWeekend*), THEN (*TurnOffAllElectricitySwitches(NataliasAppartment)*)).

2. IF (*Natalia hasPhone P*) AND (*Natalia hasCalendar C*) AND *(P isInLocation L*) AND (*L isVeryNearTo JohansOffice*), THEN (*StartAudioRecording(P)* AND *TranscribeMeetingAgenda (P, C)*)

These rules follow the Mamdani structure and can be mapped to a set of statements in a fuzzy KB as a fuzzy control system [33]. For instance, for Rules 1 and 2 we would have the mapping to fuzzy axioms in Table 2.3.

We chose *fuzzyDL* because it supports important features for expressing imprecise common knowledge when users model knowledge in SS. *fuzzyDL* provides fuzzy rough set reasoning and fuzzy reasoning for fuzzy $\mathcal{SHIF}$, which includes concrete fuzzy concepts ($\mathcal{ALC}$) augmented with transitive roles, a role hierarchy, inverse, reflexive, symmetric roles, functional roles, and explicit definition of fuzzy sets. We believe that letting end-users express domain-specific knowledge by allowing imprecise terms, can bring technology closer to them and thus, it can be better exploited.

2.2.3 Overall framework integration and implementation

Once described, in last subsections, the main components of the reactive context-aware SS architecture, we proceed to study its integration within an event based hybrid rule-based system. Figure 2.4 shows the structure and main processing modules of the fuzzy-crisp overall architecture, as well as the information flow.

The first module, *Rule Parser*, takes as input a Mamdani format IF-THEN rule's antecedent and extracts a set of (ontologically correspondent) RDF triples. These will be the event triple patterns to be subscribed to (for modification-awareness) when executing the equivalent subscription. The second module is called *Subscriber* and takes as input the RDF triples produced by the Mamdani Rule Parser, as well as the consequent of the rule. The consequent represents the actions to be performed every time the subscription's triple pattern is inserted, removed or updated in any of the KBs. The *Subscriber* then creates a subscription as output (either SPARQL or RDF based).

When an event notification is received, the consequent of the rule is to be updated, in both KBs, to keep consistency. In the case of having a fuzzy term in the antecedent of the rule, an explicit fuzzy query needs to be executed from the subscription handler method -*subscriptHandler* in Table 2.4-.

% Concrete features (Classes and Relations)

(instance Natalia Person)

(define-concept NataliasAppartment (and HousingProperty (some rentedBy Natalia))

(instance WeatherSituationTurku WeatherSituation)

(instance TurnOffAllElectricitySwitches ExecutableApplication)

(instance StartAudioRecording ExecutableApplication)

(instance TranscribeMeetingAgenda ExecutableApplication)

(functional isCurrently)

(functional hasStatus)

% Labels for the variables

VeryStormy = triangular (50,100,150)

% **A) Definition of Logical Rules as Mamdani rules**

(define-concept Rule1 = (g-and (Natalia (some hasStatus AwayForWeekend)) (WeatherSituationTurku (some isCurrently VeryStormy)) (TurnOffAllElectricitySwitches (some withParams NataliasAppartment))))

% Encoding of Mamdami Rule Base

(define-concept MamdaniRuleBase (g-or Rule1 (...) RuleN))

% Input to the controller/Facts

(instance input (and WeatherSituationTurku (some isCurrently Nearly-Cloudy)))

(instance input (and Natalia (some hasStatus AtWork))) (...)

% Defuzzification

(defuzzify-lom? MamdaniRuleBase input TurnOffAllElectricitySwitches)

% **B) Definition of Logical Rules as implication rules**

(define-concept antecedents1 (and (Natalia (some hasStatus AwayForWeekend)) (and WeatherSituationTurku (some isCurrently VeryStormy))))

(define-concept consequents1 (and (TurnOffAllElectricitySwitches (some withParams NataliasAppartment))))

(define-concept Rule1 (l-implies antecedents1 consequents1))

(define-concept antecedents2 (and (Natalia (some hasPhone P) (and (Natalia (some hasCalendar C)) (and (P (some isInLocation L))) (and L (some isVeryNearTo JohansOffice))))))

(define-concept consequents2 (and (StartAudioRecording (some withParams P)) (TranscribeMeetingAgenda (some withParams (P and C))))))

(define-concept Rule2 (l-implies (g-and antecedents2) (g-and consequents2)))

% Input to the controller/Facts

% Query for the consequent' satisfiability degree

(min-instance? input consequents1)

Table 2.3: Example: KB and rules in *fuzzyDL* for rules 1 and 2.

The types of different subscription patterns, and the correspondent *fuzzyDL* queries[4] that they origin, are shown in the mapping on Table 2.5. In this table, the subscription patterns containing *s*, *p*, and *o* represent fixed values for subject, predicate and object respectively, while *?* represents a wild-card entity. As for the query results, the entities returned will be of interest (for rule triggering) if their satisfiability degree is >0.

SPARQL query	***fuzzyDL* query**	**Subscription in M3**
SELECT DISTINCT ?user WHERE ?user mo:hasStatus mo:AwayForWeekend. ?user mo:hasName ”Natalia”xsd:string.	*(min-related? Natalia AwayForWeekend hasStatus)*	*triple = [Triple(URI(NS+”Natalia”), URI(NS+”hasStatus”), URI(NS+”AwayForWeekend”))]* *self.st = self.CreateSubscribeTransaction(self.ss_handle)* *initialResult = self.st.subscribeRDF(triple, subscriptHandler(self))*

Table 2.4: Example: Mapping of rule antecedent *”IF Natalia hasStatus AwayForWeekend”* to SPARQL and *fuzzyDL* queries

In order to test the feasibility and practicality of the proposed hybrid architecture, future benchmarking over the proof of concept is required. It can be noted that, with the technology available, the current solution assumes data redundancy, as it initially requires two (crisp and fuzzy) databases, where updates need to be twofold. Accepting this current technological drawback, we can design the experiment, where the objective is to realize a viability study of the framework. The main variable factors to consider are the time for a) Reasoning, b) Querying/Updating and c) Subscription response with respect to ontology size. For c), we account the time difference between the update of the data of interest, and the time when the notification is received. Likewise, different types of hybrid ontologies must be used, containing different proportions of both fuzzy and crisp entities. For this purpose, datasets of very large number of triples, such as the provided by the *LUBM* benchmark's data generator[5], can be used. As for the test queries, three main kinds of queries are to be considered with regard to the type of rule antecedent and consequent, which can be fuzzy, crisp,

[4] *fuzzyDL* syntax available in: `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/syntax.html`

[5] `http://swat.cse.lehigh.edu/projects/lubm/`

Subscription pattern	***fuzzyDL* query**
(?, ?, ?)	∀ *Concept C: (all-instances? C)*
(s, ?, ?)	*If s is a Concept: (min-sat? s)* *If Individual s ∈ Concept C: (min-instance? s C)*
(?, p, ?)	*If D is p's Domain and R is p's Range;* ∀ *Individual d ∈ D and* ∀ *Individual r ∈ R: (min-related? d r p)*
(?, ?, o)	*If o is a Concept: (min-sat? o)* *If Individual o ∈ Concept C: (min-instance? o C)*
(s, p, ?)	*If R ∈ p.Range:* ∀ *Individual i ∈ R: (min-related? s i p)*
(?, p, o)	*If D ∈ p.Domain:* ∀ *Individual i ∈ D: (min-related? i o p)*
(s, ?, o)	∀ *Role r, (min-related? s o r)*
(s, p, o)	*(min-related? s o p)*

Table 2.5: Mapping of subscription types to *fuzzyDL* queries

or hybrid (i.e., involving triples with crisp and fuzzy entities). Rules with different order of performance needs, are to be studied. Different kinds of rule antecedent translate into different implementation of subscription. In the case of existence of a fuzzy entity, explicit polling queries are executed in the *fuzzyDL* reasoner every time this is updated. For these cases, the crisp RDF store' subscription capability is used.

As commented before, in an *IF(x) THEN(y)* rule, *x* and *y* can contain RDF triples with only crisp, only fuzzy or hybrid terms. Due to the disparity on both KBs' capabilities and content, tasks involving different kind of rule antecedent will result having different performance. This can be due, e.g., to the fact that fuzzy rules can require more computing resources, because of the explicit continuous querying required if *manual* subscription is implemented, or due to the use of specialized semantics. However, having both crisp and fuzzy KBs can be used as an advantage for optimizing the execution time of different types of queries and datasets.

To show the equivalence among SPARQL and *fuzzyDL* queries and a subscription in *M3*, we present an example. Let us assume the user wants to add the following rule to the KB: *"IF Natalia hasStatus AwayForWeekend, THEN TurnOffAllElectricitySwitches"*. Table 2.4 shows the expressions for the equivalent mapped queries to be executed in both crisp (SPARQL) and fuzzy (*fuzzyDL*) KBs. Note that the *fuzzyDL* expression in Table 2.4 can be formulated in different ways depending on the rule's triggering criteria that best fulfils the application's needs. Another option could be querying the min. or max. satisfiability degree of the IF condition and set the triggering of our rule when, e.g., it has a satisfiability degree of min. 0.8. This is an example on how fuzzy reasoning provides more flexible or loose querying.

We identified some technical inconvenience in our proposal. The programming languages of crisp and fuzzy systems do not coincide at the moment of writing (Python and Java respectively). However, both *M3* and *fuzzyDL* are under continuous development, and a Java Knowledge Processor Interface for *M3*-Redland is expected to fully support SPARQL-based subscriptions. Therefore, a complete realization of the described experiment is part of future work.

An alternative to our proposed hybrid system could be implementing subscriptions within the fuzzy reasoner itself, as well as supporting SPARQL fuzzy querying by providing a crisp-to-fuzzy mapping. Then, issues on maintaining crisp or fuzzy semantics arise. This option would suppose extra engineering. Thus, we proposed a first basic approach, that keeps both architectures, to provide benefits from both crisp and fuzzy paradigms. In this way, the system is optimized to avoid continuous querying for changes when it is not necessary (i.e., when rule conditions are fully crisp and, in some cases, hybrid). This implementation avoids computationally expensive approaches such as continuous polling/querying, or fuzzy discretization-based solutions such as the one that *DeLorean* [32] employs. The latter results on an exponential growth of data. However, rules with hybrid antecedent (i.e., with crisp and fuzzy terms) can be generalized by setting their subscription condition to a set of *semantically wider* crisp entities, reducing in this way, the number of explicit queries to the fuzzy reasoner, any time there is a change. For instance, hybrid antecedent conditions represented by RDF triples such as *(Natalia, isVeryNearTo, JohansOffice)* can be mapped to a semantically more general subscription, formed by a crisp-only pattern: *(Natalia, isNearTo, JohansOffice).* Additionally, there can be cases where strict semantics are to be preserved, but we want to balance it with query efficiency. In this case, the condition *(WeatherSituationTurku, isCurrently, VeryStormy*) can be mapped to create a subscription for *(WeatherSituation-Turku, isCurrently, ?).* Therefore, this hybrid architecture' strategy allows for loosening of either semantics or efficiency, depending on the needs.

This section's main contribution consisted of the proposal of a hybrid architecture with a common interface that does not only support a quantitative view of SS, with crisp (SPARQL) queries and event-based rules, but also provides a qualitative factor that takes advantage of fuzzy reasoning's expressive power to handle imprecise knowledge and rules, i.e., queries with imprecise expressions or with higher complexity, abstraction or semantic levels. Such an integrated framework can be applied in a wide range of domains, from monitoring or automating activities in assisted living or e-health, to home automation or industry processes. Next section will build upon the proposed development architecture to provide a higher abstraction layer interface for end-users.

2.3 A visual language to configure the behaviour of Smart Spaces and improve interoperability and usability

The architectural needs in order to have a functional and versatile semantic framework supporting uncertainty were described in last section. In this section we propose to use Semantic Web principles of interoperability and flexibility to build an end-user graphical model for rapid prototyping of Smart Spaces applications [64, 73]. This approach is implemented as a visual rule-based system that can be mapped into SPARQL queries. In addition, we add support to represent imprecise and fuzzy knowledge by taking into use last section hybrid architecture. In this case not even programming knowledge is necessary in order to allow the user to be in control of the environment, so that it responds to context-aware changes.

2.3.1 Mapping SPARQL query language to the end-user graphical model

The survey of programming environments for novice programmers [118] shows how to lower the barriers to programming, which is one of our main aims; let non expert users to take part in the configuration of a Smart Space. Some good examples of end-user visual editors that simplify the tasks to the user when creating their own services or applications, through simple rules, are *If This Then That*[6] for online social services, *Twine*[7] for applications based on sensor interaction or Valpas [177] intelligent environment for assisted living.

A great power of visual languages is their ability of categorizations of certain primitives, and the graphical properties, to carry semantic information. Furthermore, elements of their syntax can intrinsically carry semantic information. To develop an effective visual language, i.e., one that can be easily and readily interpreted and manipulated by the human reader, we followed guidelines for visual language design [89, 149]. These can be summarized as *morphology as types, properties of graphical elements, matching semantics to syntax, extrinsic imposition of structure and pragmatics for diagrams.* In our UI, we attach meaning to the components of the language both naturally (by exploiting intrinsic graphical properties such as keeping the underlying RDF graph structure) and intuitively [89](taking consideration of human cognition, e.g. using colour to distinguish literals from classes). e.g., we considered the primary properties of graphical objects [90] to design our language's construct symbols.

[6] http://ifttt.com

[7] http://supermechanical.com/twine/

Through structuring the edition of applications as simple IF-THEN rule statements, and by using an underlying graph-based graphical structure, an end-user can model semantic behaviour, by means of classes, individuals and relationships. The RDF store, which reflects its content on the left side of the UI, shows only legal relationships and properties associated to each entity. Simple SPARQL queries can extract the required data to be presented in each view, each moment the user hovers a specific entity or menu. e.g., given a class, show its object properties associated. For example, to get the object properties of the class *GenericUser*, the following query would return *hasCalendar, worksForProject, performsActivity*, etc.

```
SELECT DISTINCT ?pred
WHERE { ?pred rdfs:domain ha:GenericUser.
    ?pred rdfs:range ?object.
    ?object a owl:Class.}
```

Listing 2.3: Query to show the Object Properties associated to a given Class

A graphical interface for representing, visualizing and interacting with SSs information is proposed to allow any end-user to model his own applications without knowledge of programming. Data gathering is possible by aggregation of different ontologies and datasets.

The graph-based and "puzzle"-like pieces to edit rules with take inspiration from the popular *Scratch* framework [176]. Variable bindings are correct, by construction of the user interface, through letting the user allocate pieces only in the positions in which corresponding data ranges and domains are allowed.

This intuitive way of expressing a rule's condition, by dragging and joining compatible (data type-wise) nodes and arcs, can be easily translated into SPARQL query patterns (e.g., conditions in the *WHERE* field) and allow fast formulation of mash-up applications. Table 2.6 summarizes the mapping applied to transform end-user visual model representations into OWL 2 entities.

The interface is based on simple IF-THEN rules applied to graph-based data. A node can be of two types, representing an OWL class (*Entity*, large and white) or a data property value (small and purple). An arc can represent a data property or object property, depending on the type of the destination node (*destNode*={Class or Value}). The THEN clause of the rule serves to 1) add, remove or update information in form of arcs and nodes (representing RDF triples) from the knowledge base, or 2) execute a registered browser-based application (with associated service grounding), possibly using concrete and well defined individuals or properties described in the IF clause or Linked Data. Registered web or Linked Data services are represented in large grey nodes. Subgraphs in IF and THEN clauses can be connected with logical operators and included into loops expressed in the

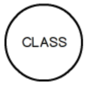
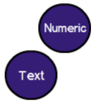
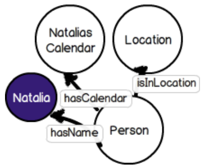
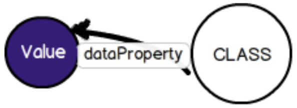
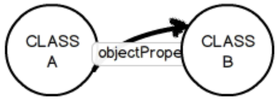
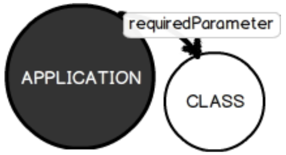
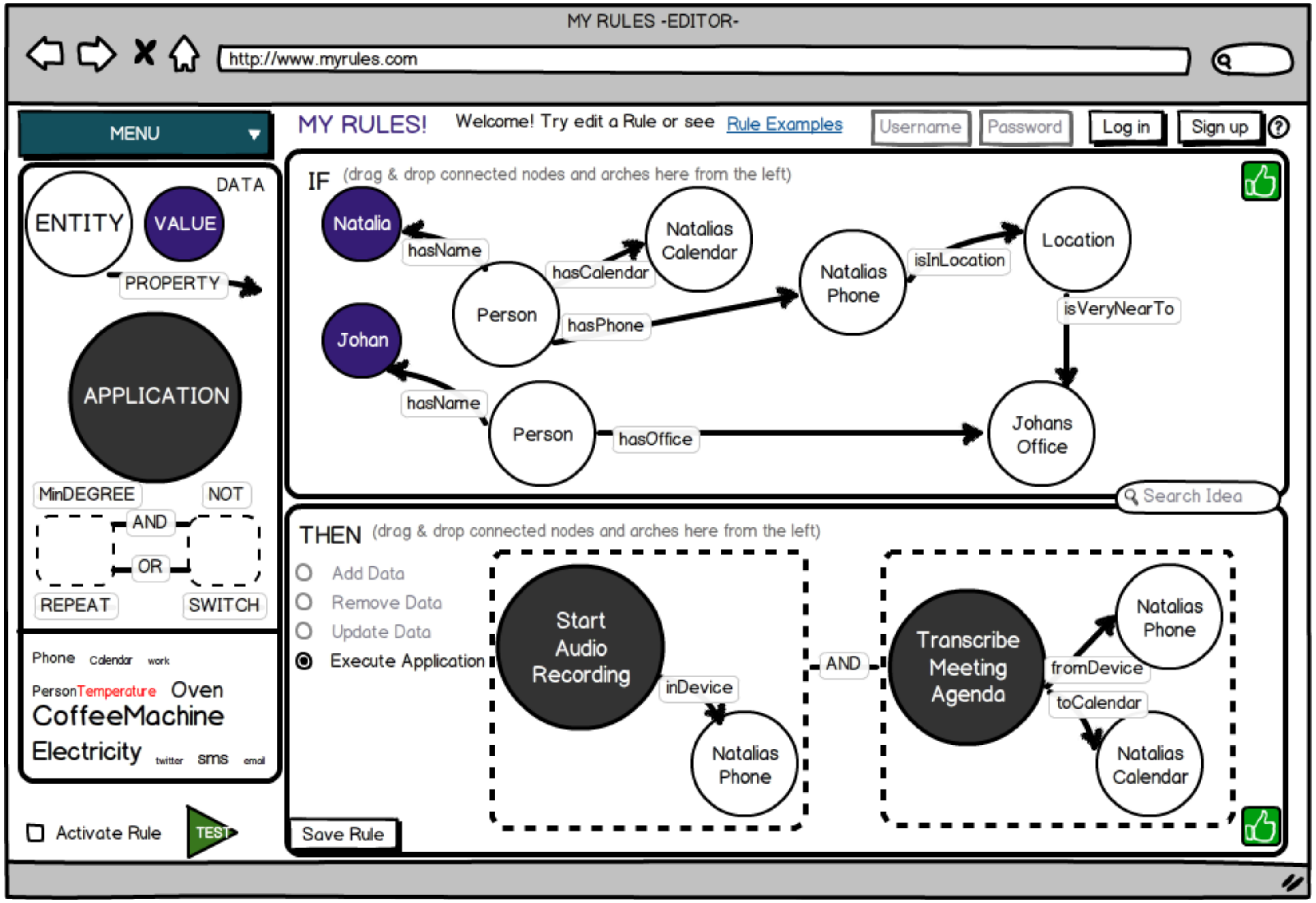
OWL2	**Appearance**	**OWL2**	**Appearance**	**OWL2**	**Appearance**
Class		Data type		Individual	 Natalias Calendar Location isInLocation Natalia hasCalendar hasName Person
Data Property		Object Property		Application/ Service	

Table 2.6: Graph visual model representation mapping to OWL2

rule's consequent. A minimum degree of satisfiability can be expressed for a determined subgraph, since a rule can be mapped to a Mamdani rule in a fuzzy reasoner (e.g. *fuzzyDL* [33]). Fuzzy modifiers are considered in the same way as crisp properties (e.g. *isVeryNearTo* in Fig. 2.5).



MY RULES! Welcome! Try edit a Rule or see Rule Examples

IF (drag & drop connected nodes and arches here from the left)

THEN (drag & drop connected nodes and arches here from the left)

Figure 2.5: User interface mock-up and example of semantic rule construction.

The mapping that transforms a graphical rule into a SPARQL query is below:

```
Initialize counter for ClassNode variables, n to 0.
Initialize processedNodes dictionary to empty.
<-IF CLAUSE MAPPING->
For each ClassNode in IFClause of the Rule:
  For each Arc leaving from ClassNode:
    If destNode is a Datatype: // Data Property Triple
      Add patterns (?indiv_n a ClassName) and
           (?indiv_n dataProp destNodeDataValue) to WHERE
      Add originNode and its index n to processedNodes
      Increment variable index n
    Else: // The Triple represents an Object Property
      If originNode is processed, obtain its index x
        If destNode is processed, get its index z
          Add pattern (?indiv_x objectProp ?indiv_z) to WHERE
        Else:
          Add pattern (?indiv_x objectProp ?indiv_n) to WHERE
          Add destNode and its index n to processedNodes
          Increment variable index n
      Else:
        If destNode is processed, get its index y
          Add pattern(?indiv_n objectProp ?indiv_y) to WHERE
          Add originNode and its index n to processedNodes
          Increment variable index n
        Else:
          Add pattern (?indiv_n objectProp ?indiv_n+1) to WHERE
          Add originNode and destNode to processedNodes
          Increment variable index n by 2
<-THEN CLAUSE MAPPING->
If THENClause.type is APP: // Execute external App
  For each ClassNode in THENClause:
    If ClassNode is processed, obtain its index w &
         add '?indiv_w' to SELECT
    Else: "ERROR: Class Nodes in APP parameters need to be defined in IFClause". Exit
  QueryResult = Run SPARQL Query with {SELECT, WHERE}
  Execute set of AppNodes with QueryResult as parameters
Else:
  If THENClause.type is ADD: // Add triples
    For each Arc marked toAdd, add pattern to INSERT
  Else:
```

```
40     If THENClause.type is REMOVE: // Remove triples
41       For each Arc marked toDelete, add pattern to DELETE
42   Run SPARQL query including {SELECT, WHERE, INSERT, DELETE}
```

Listing 2.4: Visual Rule to SPARQL Query Mapping Algorithm

The algorithm "parses" first the IF, followed by the THEN clause in the graphical model, to finally run a SPARQL query with the parameters collected in some of the array structures for SELECT, INSERT, DELETE and WHERE. Each *(origNode, Arc, destNode)* structure in the visual model corresponds to a triple pattern (subject, predicate, object). A counter n keeps track of node indexes to keep unique naming for each variable associated in the SPARQL query. Every arc is processed and, depending on the type of its destination node (line 6 & 11), the pattern is modelled as a)an individual's data property or b)an object property pattern. Generated patterns are added to the WHERE field of the query. For arcs and nodes in the THEN clause, the visual model's equivalent triple patterns are added to the INSERT or DELETE fields of the SPARQL query, respectively, since these are triples that must be marked as *toAdd, toRemove* or *toUpdate.* Finally, in THEN clause, some application (grey) nodes may require input parameters, that can reuse information from entities declared in the IF clause.

Rule Example Scenario: To study the viability of the ubiquitous model, we propose a location and context-aware scenario where positioning sensors are available through, e.g., each person's phone. We developed a Human Activity ontology that models different kind of users, their interactions and the activities they perform on the environment. Let us suppose the end-user wants to create a rule which allows her to start recording audio of the weekly meeting with his supervisor, automatically when she gets into his room: "If *Natalia* enters the room of his supervisor *Johan*, start audio-recording the meeting agenda in her phone's calendar". The aim would be keeping track, for future reference, of the agenda points and brainstorming ideas discussed, on the user's calendar. First, the user would select, from the GUI left menu the necessary entities, the datatype values for identifying the individuals *Johan* and *Natalia*, and the relations which connect these with each other. The query produced by our algorithm is below. Although 4 lines longer, it is equivalent to a straightforward query written by somebody with knowledge of SPARQL:

```
SELECT  ?calendar1 ?phone2
WHERE{ ?user0 a ha:User.
     ?user0 ha:hasName "Natalia"^^xsd:string.
     ?user0 ha:hasCalendar ?calendar1.
     ?user0 ha:hasPhone ?phone2.
     ?user0 ha:isInLocation ?location3.
```

```
    ?phone2 ha:isInLocation ?location3.
    ?location3 ha:isNear ?office4.
    ?user5 a ha:User.
    ?user5 ha:hasName "Johan"^^xsd:string.
    ?user5 ha:hasOffice ?office4.}
```

Listing 2.5: SPARQL Rule obtained from the Graphical Rule by the Mapping Algorithm

This chapter has presented architectural and infrastructure solutions in order to facilitate the programmability and interoperability of Smart Space applications. These include reconfigurability tools for both developers and non-technical end-users. After providing the user the ability to program the environment, we need to understand properly how to recognize human activities in the space for the system to react accordingly. The rest of this thesis focuses on activity recognition in order to improve understanding of the environment and to put into deployment the already provided tools for configuring and programming the behaviour of the Smart Space.

Chapter 3

State-of-the-art: Human activity recognition in Ambient Intelligence

Little semantics goes a long way

James A. Hendler, 1997

Activity recognition techniques can be divided into data-driven methods and knowledge-based approaches. Traditionally, techniques for activity recognition have focused on the branch of pattern recognition and machine learning and belong to the first group. These techniques have been extensively studied in the last decade; however, since they have been tackled from the pattern recognition perspective, they have not integrated mechanisms for semantic treatment or management. Despite this fact, they have meant an important step forward in the AmI discipline, and therefore, we devote Section 3.1 to their study. After that, a review on knowledge-driven and hybrid approaches is provided in Section 3.2, to describe those methods for activity inference and recognition from the knowledge engineering point of view. This section also includes background information on the most recent and popular knowledge engineering methodologies, i.e., semantics, the Semantic Web, ontologies, and description logics. We focus on evaluating existing ontologies that facilitate activity modelling and recognition, as well as ontological approaches to AR. Finally, Section 3.3 provides a taxonomy to evaluate all existing approaches and open challenges.

3.1 Data-driven approaches for activity recognition

Approaches for the recognition of human activities and the detection of anomalies during their performance use the information provided by sensors to build, infer, or calibrate a behaviour model. Machine learning techniques have been extensively used with this purpose, and, more specially, probabilistic models, data mining, and inductive learning. Figure 3.1 shows the concept of the activity recognition task in summary.

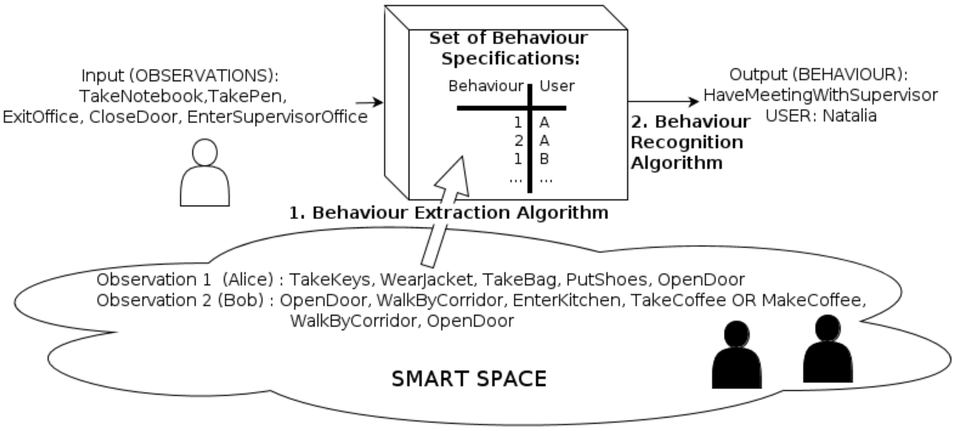

Set of Behaviour Specifications:
Behaviour User
1 A
2 A
1 B
... ...
Input (OBSERVATIONS):
TakeNotebook,TakePen,
ExitOffice, CloseDoor, EnterSupervisorOffice
Output (BEHAVIOUR):
HaveMeetingWithSupervisor
USER: Natalia
2. Behaviour Recognition Algorithm
1. Behaviour Extraction Algorithm
Observation 1 (Alice) : TakeKeys, WearJacket, TakeBag, PutShoes, OpenDoor
Observation 2 (Bob) : OpenDoor, WalkByCorridor, EnterKitchen, TakeCoffee OR MakeCoffee, WalkByCorridor, OpenDoor
SMART SPACE

Figure 3.1: Behaviour recognition in the context of Smart Spaces

Probabilistic systems provide great flexibility when controlling different alternatives in the performance of behaviours and may be easily adapted to different environments. The work [122] collects a study on advantages and drawbacks on the use of stochastic techniques for human activity recognition. Although Bayesian Networks [212, 171], Naîve Bayes classifiers [207], or non-parametric Bayesian clustering methods [210] have been used in several cases, one of the most common approaches is Hidden Markov Models (HMM) [35, 147, 231]. The systems implementing these approaches usually build a model for each activity to be recognized. These activities are checked in parallel as the sensors cast events in order to find the most likely model that fits the current human behaviour. These approaches have the power to be noise tolerant for sensor data and are capable to model sensor failure probability, but they have two main limitations: Firstly, HMM suffer from the first order assumption; secondly, if a behaviour may be performed in several ways, it is also necessary to calibrate multiple models, one for each. Multiple Behavioural HMM (MBHMM) [153] and Conditional Random Fields [122] overcome this last limitation. The determination of which tasks are currently active, even if the activity has not been concluded by the user, is possible with these models. This is very helpful when having several ways to finish an activity correctly, or when actions within an activity can be executed in any order. Another work dealing with the loss of sensor event data is [154], where they use a size and relevance-based Hierarchy of Activities of Daily Living.

Traditionally, probabilistic methods have been the most used models, but data mining techniques are also found for behaviour mining [171, 219]. Other proposed machine learning technique is inductive learning [146, 69] to mine the most frequent actions of a behaviour and their temporal relationship and to build decision trees that represent the ways in which an activity could be performed. These approaches provide a simple representation and a fast detection of the human behaviour; however, their limitation arises when activities with cyclic actions should be modelled. Regarding other models, in [79], a neural network is designed to receive data from active sensors (acceleration, temperature, etc.), which are used to infer if the user is rowing, biking, playing football, walking, running, sitting, or hiking. In this case, the noise tolerance and low computation requirements to detect the activity in real time are key points of the approach. However, due to the inherent features of neural networks, their main limitations are the difficulty to train the network with no local optima, its later adaptation to changes in the behaviour, the validation of the results, and the interpretation of the network performance.

All previous approaches use passive sensors located in the environment or active wearable sensors to acquire data. One recent trend in human activity analysis is to use computer vision to avoid body markers. An approach-based taxonomy was designed in [14] to categorize works in this area, distinguishing among non-hierarchical approaches, developed for the recognition of gestures and actions, and hierarchical approaches for high-level interaction analysis. A survey about these techniques may be found in [163, 43]. The main potential of these methods is that the information gathered by video sensors may provide much more information than passive sensors about the state of the user, its position, and its movements. In contrast to wearable sensors, these have the advantage to be much more comfortable and invisible to the user. However, the cost of these systems, the privacy loss, and the complexity of techniques to identify every action are limitations to be solved by these approaches nowadays.

Due to the growing interest in designing smart environments that reason about residents [56], intensive data collection is becoming more common. A great number of applications are tested on single user setting (naîve Bayes classifiers [212], decision trees [146], or conditional random fields [211]). Nowadays, open challenges are how to deal with multiple resident settings, interleaved activities, or social interaction. We can find preliminary approaches in the literature [196, 55, 58, 136]. For example, [196] focuses on real-time recognition of interrupted and interleaved activities, among multiple residents in smart environments. Manually labelled data from 40 residents were used to identify the most likely sequence of actions corresponding to a series of sensor events. The average accuracy was 60.60%. Parallel activities and those in cooperation have not been considered, since a strong constraint is the assumption of knowing the person's ID for each sensor event. With respect to handling social interaction, in [55], an unsupervised learning algorithm is applied to detect social interaction and monitor ADLs in the *CASAS* smart environment [171]. Activity and event density maps visualize sensor events for 15 days in a 2 resident apartment. They applied a supervised learning algorithm with two HMM (resident identifier and activity identifier models) with an accuracy of 90%. However, since not all interactions can be determined by physical proximity, they suggest to fuse the resident identification and the activity identifier models into a multi-layer hierarchical model to improve the activity recognition task in multi-resident settings.

Considering the adaptation of behaviours to changes in user routines, HMM and similar stochastic models have the limitation of being static; therefore, they cannot be applied to dynamic environments. A possible solution to overcome this limitation while maintaining the probabilistic nature of the system is [183], where human activity modelling is enhanced with the online adaptation to habit changes, environmental variations, and temporal information. In this approach, actions are ordered into temporal execution levels, and learning automata are in charge of changing the temporal execution level associated to an action as the user performs changes in his/her routine. However, the main limitation of this technique is that it is not applicable to activities with an unknown number of cyclic execution times.

Another important issues are system scalability and tackling the presence of multiple users. Approaches that address these problems can be found in [58, 55]. Likewise, it is important to consider the migration of learnt behaviours to other spaces and their adaptation to other users [172]. All these works assume probabilistic models such as HMM or Bayesian networks: A model is in charge of the inference of the user that is performing a behaviour while a secondary one performs the activity recognition. In spite of these advances, there is no general solution that integrates all the aspects desired in an SS, i.e. scalability, multi-user support, adaptation to other users or routine changes, and support for interleaved activities and social behaviours. In addition, there is no consensus on what kind of sensors should be used, what activity models or what information treatment methodologies should be employed in each case. This is due to the fact that it mostly depends on the type of problem domain and task to be solved.

However, despite the current open challenges and the limitations of the existing data-driven approaches, the best strength of these models is that they are able to handle noise, uncertainty, and incomplete sensor data [48]. In addition, they have proven to be accurate in different domains where semantics are not key, and according to different problem constraints.

3.2 Knowledge-driven approaches for activity recognition

Among knowledge-driven approaches to recognize human behaviour, we can find event calculus [129] -describing agents and actions with durations and temporal relationships- and situation calculus [135] -a logic-based framework for defining actions and changes in the representation of the world-. Other variants [47] model temporal characterization of activities and causality relationships between activities and events. The problems found in these approaches to dynamic pervasive computing environments are related with interoperability and adaptation to different scenarios, since context data sources are dynamic and not known in advance. To solve this limitation, DAML+OIL and OWL 1 ontology languages are used to formally specify context data semantics and share it among heterogeneous entities. An approach based exclusively on ontology reasoning [48] uses ontologies to represent activities as well as each data source that can be used to recognize them, from sensors to actors. Coarse-grained activities are recognized by ontological reasoning based on the available data and refined as new information becomes available. However, OWL 1 operators have not enough expressiveness to define complex relationships and tight integrations of OWL with expressive rule-based languages, such as SWRL (Semantic Web Rule Language) [swr], lead to undecidability [179]. OWL 2 treats these problems by allowing rule-based activity definitions with ontological axioms and preserving decidability, with the added advantage of providing unique semantics [179].

In [78], upper and lower level ontologies are presented for a modelling context. The lower sub-classification indicates domain dependant views of context (hospital, home, car, campus), while generic context entities include, e.g., person, device, network, physical environment, activity, location, and service. The RDF/OWL reification principle[1] is used to represent additional context attributes to the basic context triple. By means of the separation of static and dynamic contexts, context semantics and (ontology based and user defined) rules, they focus on limited resource devices. Their query engine platform, however, relies on RDQL, an RDF Data Query Language prior to the standard SPARQL. In general, a wide expressiveness model is required to deal with the description of all possible features of a user and the

[1] A reified RDF data contains each original statement as a resource and the other additional statements made about it

functionalities provided by devices and systems [148]. For instance, in [148], an ontology models Ambient Assisted Living (AAL) services for the elderly in a domotic domain. Their particular aim is to facilitate the validation of accessibility (i.e., disability constraints) for users.

Other models, based on fuzzy logic [13], have been also developed. For instance, in [93], embedded agents are connected to sensors and effectors and use fuzzy logic-based *Incremental Synchronous Learning* to define particularized rules instead of seeking to extract generalized rules. Based on rule bases built from previous (e.g., user or environment) occupiers, the learning time is minimized, since the system starts from a similar rule base to later refine the rules. Another approach in the same way uses agents to discretely control the Essex intelligent dormitory iDorm [74] after an adaptive learning of human behaviour. Agents use appliance parameters as input to a fuzzy logic controller acting over effectors. Through a fuzzy markup language (FML), a detailed structure of fuzzy control can be defined independently of its legacy representation, allowing agents to capture user habits and to apply an adaptive rule-based strategy.

Hybrid approaches, i.e., systems that combine data-driven and knowledge-driven approaches for activity recognition, are found in works such as evidential network-based activity inference [101] or COSAR [178]. Ontological reasoning with OWL 2 is used to recognize complex activities based on elementary sensor data and simple activities recognized through data-driven methods (in this case, statistical reasoning) [179]. COSAR is used together with the PalSPOT ontology, within the context aggregation middleware CARE [15]. The COSAR system retrieves information on simple human activities using hybrid ontological/statistical reasoners. They show how the recognition rate improves considerably as well as how the error rate is reduced by 45.43% with respect to the uniquely statistical technique. COSAR avoids miss-classifications between activities characterized by similar body movements but different contexts. One disadvantage of this approach, however, is that they use location association to infer activities and therefore, this can be a problem when recognizing fine grained activities that occur in a unique, confined, or small size space.

Another example of a hybrid approach combines ontology-based context reasoning together with computer vision research and integrates a scene tracking system with an ontological layer to avoid limitations that make classical object tracking procedures fail in complex scenarios [82]. Abductive and deductive reasoning is used to build different abstraction levels. The scene interpretation serves to generate feedback to the quantitative tracking procedure. In this way, DL reasoning reduces the complexity in the implementations for scene interpretation. One of the challenges, however, is not only the definition of suitable ontologies, but also the detailed creation of abduction rules.

3.2.1 Ontology-based and hybrid approaches for activity recognition

Ontologies have shown to be useful in hybrid approaches to context reasoning, among others, in computer vision research where 2D scene tracking is integrated with ontological layers to avoid issues that make classical object tracking procedures fail in complex scenarios [82].

Recent works show that hybrid approaches are promising, for instance, [224], where they model spatio-temporal scenarios with non-ontological fuzzy semantics, or [227], a concurrent activity recognition KCAR (crisp) ontological approach focused into concurrent multi-user activity recognition. However, the validation is done with a dataset where activities are discriminable by location characterization, and where all data is discretely labelled (from different, but mainly positioning sensors). In our framework we handle both continuous and discrete streams of data.

Other present approach is [49], where ontological reasoning is used for non-concurrent real-time continuous recognition of fine and coarse-grained single-user activities. However, although it includes partial uncertainty (handling simulated faulty-sensors and changes in objects used), it does not combine domain knowledge with data-driven approaches nor tackles a continuous input dataset. A 94.44 % average activity recognition rate was achieved with an average recognition runtime of 2.5 seconds.

Saguna et al. [188] propose a probabilistic Markov chain analysis to discover representative activities through activity signatures that serve to generate complex activity definitions. They achieve high accuracy for concurrent and interleaved activities. Despite using a hybrid and unifying theoretical framework, exploiting both domain knowledge and data-driven observations based on *Context-Driven Activity Theory (CDAT)*, the dataset employed is only discrete and synthetic. Furthermore, although it is a semantic approach, they do not use an ontology to fully exploit automatic knowledge inference nor uncertainty reasoning.

Okeyo et al. [156] use a knowledge-driven method where an ADL ontology and a dynamic sensor data segmentation method [158], based on shrinking and expanding the time window, produce high-accuracy recognition of daily activities. Their approach provides very good ad-hoc results under a synthetic dataset; however, they do not provide results under realistic scenarios, which are always more complex and have an inherent component of uncertainty. On the other hand, their system does not consider movement tracking for sub-activities and their approach is evaluated with a discrete input stream as a whole, which does not always occur in practice, due to the heterogeneity of sensors for data acquisition.

A common lacking element found in existing hybrid and ontological AR systems is the support for modelling uncertain, vague, and imprecise information [71, 61], which is an inherent feature of activity recognition, and that can provide meaning to lower level data. Furthermore, there are also needs for hybrid AR systems which can handle the problems of both low-level activity detection in real time, as well as the semantics, context-awareness and uncertainty typical of high-level activities. Uncertain or vague data should be used as a natural way to provide flexibility to the model, for it to adapt to real life situations. In Chapter 6, a solution to a hybrid system that treat these problems, will be presented.

3.3 Taxonomy on approaches to human activity recognition

The state-of-the-art on human behaviour monitoring and recognition from previous sections can be summarized through a taxonomy that considers the different aspects this thesis handles. Taking into account the approaches studied, the taxonomy in Figures 3.2 and 3.3 establish the assessment criteria to evaluate human activity modelling. Each criterion and its correspondent key aspects are specified below, together with a list of publications that tackle those criteria and key aspects. These publications are examples selected from previous sections as the most representative ones in their respective category.

This classification allows us to summarize the existing methods in the literature for human activity inference. In addition, it enables the classification of a human activity recognition system in a multi-modal and taxonomic way, according to the methodology, special needs to model, and application domain considered. Next, we provide an evaluation of the mentioned techniques. We mainly focus on data-driven approaches, since knowledge-based ones, and more specifically ontologies, are analysed in the following sections.

The first criterion, *learning procedure*, distinguishes among data-driven, knowledge-based, and hybrid approaches. Regarding data-driven methods, we notice that most of the proposals are based on supervised learning. On the other hand, ontological models are the most frequent methods used in knowledge-based techniques. The main difference between them, as stated in previous sections, is the inclusion of context-awareness tools to include semantics in knowledge-based approaches. However, data-driven cases have provided very promising results for the inference and recognition of human activities in the environments where they have been tested. Unsupervised data-driven methods should be highlighted here, because their nature about not needing any previous training makes them suitable for the adaptability of systems to other environments, they may contribute to the invisibility of the ubiquitous system to the user and to reduce the interaction between humans and environment. However, in our opinion, these approaches are

Figure 3.2: Taxonomy for Human Activity Recognition (Part I)

Taxonomy for Human Activity Recognition (Part I)

- *CRITERIA 1*: LEARNING PROCEDURE
  - Data-driven approaches
    - Supervised
      - Inductive ([69, 146])
      - Statistical ([30, 175, 221])
      - Reinforcement learning ([183])
    - Unsupervised ([55, 136, 141])
  - Knowledge-based approaches
    - Logic-based approaches ([129, 135])
    - Ontological ([180, 30, 175, 157, 82, 48, 50, 229, 179])
    - Rule-based systems ([93, 13])
  - Hybrid ([30, 178, 82])
- *CRITERIA 2*: TECHNIQUE/METHOD
  - Graphical models:
    - HMM ([196, 35, 231])
    - MBHMM ([153])
    - Bayesian Network ([212, 207])
    - Clustering ([136])
    - Conditional Random Fields (CRF) ([211])
    - Emerging Patterns ([85])
    - Learning Automata ([183])
  - Non Graphical models:
    - Data Mining ([114, 171, 219, 35])
    - Neural Networks ([77, 20])
    - Naîve Bayes ([38])
    - Support Vector Machines (SVM) ([221, 38])
    - Rule-based Systems ([13, 82])
    - Fuzzy Logic ([13])
  - Hybrid ([151, 136, 141])

Figure 3.3: Taxonomy for Human Activity Recognition (Part II)

Taxonomy for Human Activity Recognition (Part II)

- *CRITERIA 3*: SOCIAL INTERACTION
  - Social interaction/Shared activities ([55, 196, 136])
  - Multi-resident setting/multi-user tracking ([186, 58])
  - Concurrent activity recognition ([85, 136])
- *CRITERIA 4*: SENSOR INFRASTRUCTURE
  - Computer vision techniques ([53, 223, 82, 139, 163, 43])
  - Passive sensor techniques ([175, 18, 26, 121, 74, 93, 69, 196])
  - Wearable sensor techniques ([138, 173, 130, 77, 151, 79])
  - Dealing with loss of sensor data ([153, 154])
- *CRITERIA 5*: SCALABILITY
  - Adaptability to changes in behaviour ([201, 157, 172, 207, 183])
  - Behaviour extensibility to other users ([171, 172])
  - Interleaved activity recognition ([196, 85])
  - Publish/subscribe infrastructure ([189, 198])

still in their first stages and much more work could be done to achieve this goal. In addition, self-adaptive supervised techniques are also under development and could provide suitable solutions for this problem.

The second criterion, *technique*, looks at the specific algorithm or model used for the recognition phase. We distinguish graphical models as those that provide a graphical representation of a behaviour. This is an important feature to achieve modularity and ensure a consistent way to build user interfaces for data accessibility [197]. However, the complexity of these models for a non-expert person makes the provision of this system feature difficult. Moreover, the model should be accurate enough to minimize the behaviour detection failures, and non-graphical methods could provide better performance in some applications. In [183], the authors suggest the separation of the graphical representation of the human activity from the learning procedure with this aim, although their approach is difficult to extend to all possible human behaviours.

Another criterion is the support to model *social activities* or human interaction, where multi-user settings or shared activity features are considered. Here, the current sensor technologies have the challenge to determine which user is performing each detected action, to model interleaved, social, and interchangeable behaviours. The approaches provided with this aim are promising, but the underlying sensor architecture of the environment is a key aspect for these techniques to succeed. In this way, the fourth criterion, *sensor infrastructure*, classifies the approaches according to the sensors used, i.e. video, passive, or wearable sensors. Recently, researches have made a great effort to develop non intrusive and accurate video sensors to overcome the previous limitations. However, the complexity and cost of these solutions are problematic in both hardware and activity modelling fields.

At last, *scalability* is a criterion that considers adaptation to changes in a routine, the accommodation of a given behaviour to other users, and the scalability of the learnt activities to other environments. In our opinion, this issue is essential to make commercial applications out of ubiquitous spaces. However, the goal of the environment, the sensor technologies required for each application, and the best inference and recognition model in each case are limitations which are difficult to be addressed. The integration of semantics within these systems as a new abstraction layer could help overcome this challenge [179, 82].

3.4 A survey on ontologies for human activity recognition

Smart Spaces are considered to be context-aware systems; therefore, a key requirement to design such systems is to give computers the ability to understand situations and environmental conditions [45]. To achieve this, contextual information should be represented suitably for machine processing and reasoning. Semantic technologies, and more specifically ontologies [86, 83], are well suited for this purpose because ontologies allow to share knowledge while minimizing redundancy. In addition, they are good tools for knowledge representation and reasoning. In this section we study the formal background to these methodologies and some practical implementations of ontologies for our purpose.

3.4.1 Semantics and the Semantic Web

Semantics is the branch of linguistics and logic concerned with meaning. Its two main areas are: a) *logical semantics*, concerned with matters such as sense, reference, pre-assumption, and implication, and b) *lexical semantics*, dealing with the analysis of word meanings and relations between them[2]. In systems equipped with semantic tools, information is given well-defined meaning so that it enables computers and people to work in cooperation.

The Semantic Web [29] paradigm was introduced as a collaboration of the W3C and others to provide a standard for defining data on the Web. The Semantic Web was defined by Tim Berners-Lee et al. in 2001 as "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation". The SW uses XML tags that conform to the Resource Description Framework (RDF) and Web Ontology Language (OWL) formats. SPARQL is the W3C standard query language for RDF (Resource Description Language) since 2008.

While data-driven approaches for activity recognition suffer from *ad-hoc* static models, data scarcity, and scalability [48], Semantic models can fulfil the needs of context-aware personalized activity-based environments where multimodal sensor technologies are constantly being introduced. Simple data systems can be modelled through key-value and markup models such as CC/PP [126], while more complex domains require more sophisticated formalisms such as object-role based models, spatial models of context, or ontologies. In general, expressiveness requirements in human behaviour and environment representation include the ability to represent hierarchical structures, complex relationships among context instances and complex definitions based on simpler ones, usually using restrictions that may be spatial or temporal. Ontologies have shown, in the literature, to be one of the most promising tools to achieve these goals. In the Semantic Web, ontologies represent the main technology for creating interoperability at a semantic level. This is achieved by creating a formal illustration of the data, making it possible to share and reuse the ontology all over the Web.

[2] Oxford Dictionaries, `http://oxforddictionaries.com/definition/english/semantics`

3.4.2 Ontologies and Description Logics

Among semantic technologies, ontologies are the most used formalism to represent and reason with knowledge. An ontology can be defined as a "formal specification of a shared conceptualization" [37]. It offers a formalism to represent classes or concepts, individuals, relations, functions, and attributes. One of the main advantages of using ontologies is their way to represent and share knowledge by using a common vocabulary. They formulate and model relationships between concepts in a given domain [59]. As providers of a format for exchanging knowledge, they promote interoperability, knowledge reuse, and information integration with automatic validation. Ontologies separate declarative and procedural knowledge, and facilitate the modularity of the knowledge base (KB)[31]. They also allow information to become not only human but also machine-readable and processable by agents. Ontologies have been used in heterogeneous problems such as intelligent m-Government emergency response services (e.g., disasters and attacks) through case-based reasoning [17] or detecting information system conflicts in requirement analysis phase [137], just to name a few.

There are three main streams of Semantic Web languages, divided in triple languages (RDF and RDFS), conceptual languages (OWL, OWL 2, and their profiles OWL EL, OWL QL, and OWL RL), and rule-based languages (such as SWRL and RIF) [203]. At the same time, OWL comprises three sub-languages of increasing expressive power: OWL Lite, OWL DL and OWL Full [33]. We focus this work on OWL 2 and OWL DL.

To be more specific in our context, let us put an example of an ontological activity modelling using the expressiveness of OWL 2 language. The activity $WalpurgisParty$ as a *friendly meeting in which all the participants are wearing a white hat*, can be written as an OWL 2 axiom as follows:

Example 1: $WalpurgisParty \sqsubseteq \mathit{FriendlyGathering} \sqcap \forall\, \mathit{hasActor}.(Person \sqcap \exists\, isWearing.WhiteHat)$
$\mathit{FriendlyGathering} \sqsubseteq \mathit{Activity} \sqcap \geq 2\, \mathit{hasActor}.Friend$

where a *FriendlyGathering* is an activity having at least two actors who are friends.

Not only the power of representation is key in ontologies. Reasoning capability is another important requirement for a knowledge-based activity recognition system. It is used to derive implicit information from explicit context data. For instance, the user's current activity can be inferred based on his/her current location, posture, used objects, and surrounding people. Reasoning can also automatically detect inconsistencies in the KB [179]. Logical reasoning consists of deduction, abduction, and subsumption, to extract a minimal set of covering models of interpretation from the activity model KB based on a set of observed actions, which could explain the observations [48]. For example, the simple statement ”Two people working in the same project and institution are colleagues” may be formulated as follows, to infer which people are colleagues in the system:

Example 2: $Person(?x) \wedge Person(?y) \wedge worksInProject(?x, ?p) \wedge worksInProject(?y, ?p) \wedge worksForInstitution(?x, ?i) \wedge worksForInstitution(?y, ?i) \rightarrow isColleagueWith(?x, y?)$.

The Web Ontology Language (OWL) is based on the knowledge representation formalism of Description Logic (DL) [25], which models concepts, roles and individuals. Description Logics (DL) are the most used languages to model formal ontologies. DL reasoning can support incremental progressive activity recognition and assistance as the activity unfolds.

In DL, the terminological box or *TBox* is the vocabulary used for defining concepts and roles within a domain, while all instances or named individuals conform assertions about a real world domain in the *ABox*, which is the assertional box. While the TBox contains sentences describing concept hierarchies (i.e., relations between concepts), the ABox contains ground sentences stating where in the hierarchy individuals belong. Statements in the TBox and ABox can be interpreted with rules and axioms in DL to enable reasoning and inference, including satisfiability, subsumption, equivalence, disjointness, and consistency. DL reasoning supports decidability, completeness, and soundness in polynomial time complexity for an inexpressive DL and in exponential time complexity for expressive DLs [25].

Ontology-based activity recognition provides a number of advantages [51]. They support incremental progressive activity recognition and facilitate course-grained and fine-grained activity inference with the possibility for data fusion and semantic reasoning [48], including activity learning, activity recognition, and activity assistance [61]. Other benefits of ontology-based AR are the ability to discriminate the significance and urgency of activities through semantic descriptions [51]. They provide state-based modelling and a robust reasoning mechanism. Since sometimes, a mathematical description of a behaviour, e.g., *morning routine*, cannot be trivially provided, ontology-based reasoning allows extra pieces of data to be used for behaviour disclosure. Through the detection of low-level events reflecting the state of each individual entity, ontology-based reasoning can assert when a task or entity is different or the same as another one. In order to achieve this, the domain-specific knowledge needs to be unambiguously defined.

When the conceptualization of activities and their interrelationships on ontological activity models encodes rich domain knowledge and heuristics in a machine understandable way, a higher degree of automation is enabled for knowledge based intelligent processing [48]. In addition, as new context sources are constantly being introduced in ubiquitous environments, data-driven approaches (e.g., supervised learning) require to re-train the complete model again, before being able to adapt to changes in the modelled activity. Furthermore, to re-build the updated model, data collection with the new context data sources is required. However, knowledge-based methods allow previous activity recognition models to be used; updating only the affected context-rules is enough to recognize the adapted activity.

Ontologies' properties such as flexibility, reasoning, information sharing, and knowledge representation make these models one of the most promising tools for the purpose of activity recognition [51]. Both the environment and the user can be provided with semantics to help in the context definition process. Moreover, including semantics aids in the management of user information and the interaction with the system, facilitates the definition and comprehension of human behaviour and thereafter, helps to develop new learning and recognition models.

Despite their advantages, a major limitation is the lack of support for imperfect information, since it is not inherent to OWL 2. Previous experience when defining complex activities has also shown some limitations, e.g., on the tree model property [179]. Adhering to this OWL 2 property guarantees decidability on reasoning problems but also limits the expressiveness, requiring every predicate (in an object property) to contain a quantified variable[3]. On the other hand, most rule-based languages do not impose such forced restrictions [179].

Ontologies for human activity representation

There is a broad variety of ontologies and vocabularies in the literature to model context in smart environments. Users are the central part, as well as what happens in their surroundings. The following ontologies show general user-centred approaches to model human activities:

- The *CoBrA-Ont* [44] ontology is an extension from SOUPA (Standard Ontologies for Ubiquitous and Pervasive Applications). It defines people, places, and activities. It was designed to reason within the CoBrA (Context Broker Architecture) infrastructure and defines key ontology categories such as action, agent, time (instant and interval), space, device, etc. One application scenario is the eBiquity group meeting ontology that models video presentations, recorded discussions, and other media material from meetings and their coordination by agents

[3] e.g., "an internal meeting is a meeting in which all actors are colleagues among themselves" is impossible to express in OWL 2 without giving up decidability. To solve this, restricting the activity definition to a specific domain is necessary, e.g., "an internal meeting of company X is a meeting in which all actors are employees of X" [179].

and actors. CoBrA specially takes care of places, distinguishing between atomic and compound places, depending on their containment property, defined as a spatial capability of subsuming other physical locations. It also distinguishes (person and software) agents, with their respective home pages, email addresses, etc. Each agent has his/her respective role, e.g., speaker or audience role, activity context, and location. CoBrA integrates a privacy policy language for user privacy protection that extends the Rei policy language[4]. User privacy is considered by restricting the sharing of contextual information acquired by hidden sensors or agents.

- The *CoDAMoS* [166] ontology defines four main core entities: user, environment, platform, and service. The aim behind this ontology design was to provide application adaptation, automatic code generation, code mobility, and generation of device-specific user interfaces. Resources are especially modelled (memory, network, power, storage resources), as well as service profiles, groundings and different kinds of software (middleware, OS, rendering engine or virtual machine). Two levels of granularity offered are tasks and activities. Users can have a mood or be located in absolute or relative locations with different environmental variables (Fig. 1.4).

- The *Delivery Context Ontology* [40] (W3C) provides a formal model of environment characteristics in which different devices interact with concrete services. This ontology includes device characteristics, the software used to access the service and the network (and network bearer) providing the connection, among others. Other entities modelled in this ontology are environment, hardware (battery, memory...), tactile input or text input types, cameras, aspect ratio, software (web browsers, script language, page markup restrictions), character set, bluetooth profiles, location, unit conversions, and physical quantities (measures from Coulomb to inches).

- *SOUPA* ontology [46] (Standard Ontology for Ubiquitous and Pervasive Applications) is divided into two main blocks called SOUPA-Core and SOUPA-Extensions. They are used in the CoBrA architecture. SOUPA-Core defines general concepts to appear in different scenarios, e.g., person, agent, policy (right, prohibition, obligation, dispensation, each of them with an associated actor and action), actions (preconditions and effects), events, geo-spatiality, space (locations' longitude, latitude, and altitude), time, and MoGATU BDI ontology (belief, desire and intention, goals, plans for agents). A policy in SOUPA is a

[4] Rei defines a set of ontology concepts for modelling rights, prohibitions, obligations, and dispensations in the domain of security

set of rules defined by a policy creator, which is to be enforced by some policy enforcer. SOUPA-Extensions support particular concepts in narrower domains (e.g., home, office, entertainment). Extension ontologies demonstrate how to expand SOUPA-Core ontologies to define a set of vocabularies that support peer-to-peer data management in pervasive computing environments. Some examples of these ontologies consider instant time and intervals, spaces ((subsumed) spatial regions, geopolitical entities -to which policies apply-), a region connection calculus ontology, meeting ontology with event (re)schedules and cancellations and user contact preferences. The *EasyMeeting* infrastructure facilitates typical user activities in meetings, such as setting up presentations, controlling services via speech, and adjusting lighting (light action control ontology) and background music, based on the state of the meeting. They also offer the priority ontology, which is established for a set of desires and intentions of an agent, and the ontology to describe conditional beliefs.

- The *mIO!* ontology [165] is a network ontology, developed using the NeOn methodology, that represents user context to configure, discover, execute, and enhance different services in which the user may be interested. The NeOn methodology basically considers the reuse, merge, matching, and re-engineering of ontological resources. Eleven modular ontologies define the mIO! core: User (groups, organizations, their employment status, skills, mobility pattern, and online identities; reuses *FOAF* ontology as a whole), Role (knowledge about profiles, preferences; reuses ontologies such as *Reco* for user preferences), Environment (environmental conditions; reused from CoDAMoS), Location (spatial entities and area and distance units from SOUPA, location coordinates, buildings, countries), Time (temporal units and entities, instants, intervals, reuses W3C Time ontology), Service (from business to mIO! services -with digital signature, input and output parameters, its components and functionalities), Provider (wide categorization of simple and aggregated service providers, from business to software services), Device (taxonomy categorization and "componency" pattern, charging mode, compatibility with standards, e.g. glucose meter, pulse oximeter, anemometer, etc.), Interface (types, I/O modalities and characteristics), Source (aggregated or not: user, device, service, etc.), and Network (communication networks, network topologies, operators and administrators, accessibility, price, coverage, etc.; network types and modes reused from *Delivery Context* ontology).

- The human activity recognition ontology [179] from *PalSPOT* project models individual and social activities. The types of interaction are modelled as acknowledgement, asking for opinion, comment, negative/positive opinion, proposal, or request of information. Activity granularity is slightly shown (basically, only one activity-level); however, an extensive taxonomy is available for personal, physical, and professional activities, travelling activities, postures, and activities using different kinds of devices or artefacts. An interval-based representation of activities models the overlapping of these in time. Other entities are indoor -corridor, arcade, elevator,...- and outdoor -promenade, sidewalk, rails...- communication routes. Symbolic locations (indoor, outdoor -pedestrian or not-) and time granularity are provided. The PalSPOT ontology is used within the context aggregation middleware CARE [15], which maps context data to ontological classes and properties, and interacts with the COSAR system [178], which retrieves information about simple human activities using hybrid ontological/statistical reasoners.

- *CONON* (CONtext ONtology)[222] defines general concepts in an upper ontology such as location (indoor, outdoor, with different environmental features and variables, weather condition), activity, person, or computational entity (such as devices with a status -phone, TV, DVD player...-). CONON allows extensions in a hierarchical way by adding domain specific concepts, where different intelligent environments are modelled (home, office, vehicle, etc.). Activities (with start and end time) are divided into deduced (dinner, movie) and scheduled (party, anniversary) activities. The status of indoor spaces entities, e.g., curtain, door, window, is also represented. Some domain-specific ontologies are the home-domain (e.g., sleeping, showering, cooking, watching TV, having dinner) and office-domain ontologies. Reasoning categories employed are DL ontology reasoning and user-defined reasoning using first-order logic (through customized rules).

- The Pervasive Information Visualization Ontology (PiVOn) is a formal context model [99] composed of four independent ontologies (users, environment, devices, and services) which describes intelligent environments. Some properties of the main elements in the User ontology are location, identity, activity, and time. The context is analysed from the perspective of the 5 Ws Theory[5] to design context-aware systems. The result can be summarized in a two-dimensional taxonomy of context elements: the first one defined by the four main categories of the context (user, environment, device, and service) and the second, by the 5 Ws. Events in the ontology have reminders, schedule, and are part of a user agenda. They involve contacts (FOAF) from the user,

[5] A journalism principle regarded as basic in information gathering (What, Who, Where, When, Why).

who can be in user situations, and possibly accompanied by some other user(s). Users perform tasks (which can have subtasks) that have a goal and use some services. Tasks have types, roles, significance levels, progress, time and space issues, and attention levels. User situations also play a role and belong to the user availability, a state of mind, or a task. The device ontology determines the types of devices (autonomous, dependent, sensor, actuator), the services provided by a device, the owner of the device, status, location, its hardware and software profiles, communication hardware profile, its use and compatibilities, etc. The environment ontology represents the co-location of objects, *near* (*inFrontOf*, *on, under, behind*), *includedIn, associated*, etc. Spaces are modelled with the area they are located in (building, wing, floor), its purpose, structure, and capacity. The visualization service ontology includes, for each service, an associated visualization service (displayed by devices) which contains content. The content has a visual form and is transformed into data. The visual form has different types of scalability parameters (filtering, pagination, complexity, latency). A prototype scenario is modelled on an academic conference.

- The Situation ontology [225] is divided into situation and context layers. Situation is defined as a set of contexts in the application over a period of time that affects future system behaviour. A context is any instantaneous, detectable, and relevant property of the environment, system, or users, such as location or available bandwidth. The ontology considers atomic and composite situations. The latter are composed of temporal, conjunction, disjunction, and negation situations. Composite situations can integrate atomic situations through Boolean context operators that act over context value domains. Similarly, a temporal situation has temporal operators over time periods. Regarding context, the Situation ontology is classified into device, user, and environment context. An entity can satisfy a situation by having related context data, with a certain context value (e.g. float temperature value), within a context domain value (e.g., available-memory context). Context value domains are provided with data context operations. An example of a smart conference scenario could specify the situation *ReadyForMeeting* as the conjunction of two atomic situations: *InConferenceRoom* and *LightOn*, where the *InConferenceRoom* situation' *location-context* value is the same as *crLocation*, and the *LightOn* situation is represented as "the *lightContext* value is true".

Ontologies for context and environment representation

We described, in previous section, a set of ontologies that focus on the user. In this section, we describe a series of very helpful concrete domain ontologies to represent the context and the environment where human activities occur:

- Location: *PlaceTime.com*[6](URIs for places and times) contains instants and intervals in the Gregorian calendar and points in the WGS 84 datum, utilizing the RDFIG Geo vocabulary. Other vocabularies useful in object and human location tracking are *WGS84 Geo Positioning*[7] or *GeoNames*[8].

- Time ontology[9], developed by the W3C Semantic Web Best Practices and Deployment Working Group (SWBPD), describes temporal content and properties of web pages and web services. It also provides topological relations among instants and intervals, durations, datetimes and world time zones.

- User profile and preferences: People can be modelled with the *FOAF* ontology[10]. User Agent Profile (UAProf)[11] specification relates capabilities and preference information for wireless devices. The *CC/PP* (Composite Capabilities/Preference Profile) model [126] is a W3C initiative that suggests an infrastructure (and vocabulary) to describe device capabilities and user preferences. The representation model can guide the adaptation of the content presented to the device, considering software terminals, hardware terminals, applications such as a browser, data types, protocols, and specification conformance of products (documents, producers, and consumers on the web). The hierarchical structure of components is divided into three areas: hardware, software, and application[12]. Also, a W3C Delivery Context Ontology and a glossary of terms for device independence exists (with discontinued maintenance). The Ontologies in the semantic desktop *Gnowsis* project focus on use cases such as tagging a file, e-mail, or a website[13]. The User Action Ontology in *Nepomuk* (Social Semantic Desktop) describes desktop events and their Calendar Ontology (NCAL) adapts the W3C ICALTZD ontology[14].

[6] http://placetime.com

[7] http://schemapedia.com/schemas/geo

[8] http://www.geonames.org/

[9] http://www.w3.org/TR/owl-time/

[10] Friend of a Friend Vocabulary: http://xmlns.com/foaf/spec/

[11] http://www.w3.org/wiki/UAProfIndex

[12] UAProf and CC/PP are encoded in RDF/S. http://www.w3.org/TR/CCPP-struct-vocab/

[13] http://gnowsis.opendfki.de/

[14] http://oscaf.sourceforge.net/ncal.html

- Online Behaviour: The *SIOC* (Semantically-Interlinked Online Communities)[15] ontology describes information from online social communities (e.g., message boards, wikis, weblogs, etc.) on the Semantic Web. A natural extension to SIOC for online behaviour is the *Open University Behaviour Ontology* (OUBO), which allows user behaviour to be captured over time and facilitates role inference in a way that a user's role in a given context can be derived through semantic rules [185].

- Content extraction: The Image Annotation W3C[16] ontology for semantic image annotation and retrieval can be used for deep multimedia analysis (e.g. image-based context recognition). *Nepomuk Multimedia Ontology* (NMM) also defines metadata for multimedia files, as well as *EXIF* ontology describes digital camera images and image management software[17].

- The ASC (Aspect-Scale-Context) model [205] includes concepts such as aspects, scales, and context information, each aggregating one or more scales. Although useful to describe measurement units, it cannot describe more abstract context information, like user activities. The DAML-based Context Ontology Language (*CoOL*), derived from the model, can be used to enable context-awareness and contextual interoperability.

3.4.3 Domain-based classification for ontology evaluation

Once the main ontologies for human behaviour representation have been described in previous section, we now provide an evaluation of all relevant design aspects to select and develop new tools to improve the construction of competitive human activity recognition systems. There exist a large number of different scenarios deployed within heterogeneous ubiquitous spaces, as well as different technologies and approaches. Thus, the diversity of systems makes reaching a consensus in evaluation tools quite difficult. A lack of evaluation tools prevents the use of a well-formed hierarchical system classification, since there is no widely accepted model to be reused in different domain applications [165].

In [27], an evaluation framework for upper ontologies on situation and context-awareness is provided. They analyse four ontologies: SAW (Situation Awareness ontology within SAWA project [145]), Situation Ontology [225], SOUPA [46], and CONON [222]. The evaluation framework focuses on top-level concepts, SAW concepts, and modelling characteristics of upper

[15] `http://rdfs.org/sioc/spec`

[16] `http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/`

[17] *Shared-Desktop-Ontologies* (SDO) `http://oscaf.sourceforge.net/`

ontologies, such as *universality* or *articulation.* Top-level concepts evaluated are object, attribute, relation and role, event, and situation; while SAW-specific concepts consider space and time, thematic roles, situation types, and situations as objects. Even though none of the ontologies fulfils all or at least most of the overall criteria, we can affirm that SAWA satisfies the modelling of most concepts, followed by Situation Ontology, SOUPA, and CONON.

Other examinations evaluate different ontologies focusing on licensing (open or free to use), maturity (showing evidence of use), structure (modularity, design), granularity of time and space, vagueness, security [193]; or availability, existence of management tools, expressiveness, etc. [113, 26].

The taxonomy proposed in [12] to evaluate a wide variety of pervasive computing systems presents some evaluation criteria that consists of the concepts *Architecture, Application Purpose, Autonomy, Integration, Interaction, Intelligence*, and *Service Availability.* The infrastructure defines the architecture and design. The application purpose describes, among other features, the quality of context, its reliability, fault tolerance, security, privacy, and effectiveness. Autonomicity considers how a pervasive computer system is initialized, how it evolves, how it automatically tackles failures adjusting to users, how it integrates new resources, and how it fends off attacks. The interaction criterion, on the other hand, identifies human to machine and machine to machine presentation capabilities. Intelligence criteria measure the pro-activeness of the AmI technique as well as the quality of context and adaptability to changes. At last, Service Availability categorizes the pervasive service based on the ability to be ”anywhere anytime”, e.g., discovery, deployment, mobility, etc.

Evaluation of ontologies for human activity recognition

Despite the fact that general ontology evaluation methods exist, none of them is focused on the field of human activity recognition. We focus on specific features related with human behaviour modelling in accordance with their environment. Our starting point is the previous study in [165], where different ontologies are compared with respect to their support for modelling different domains. Once we have made an in-depth analysis of the most relevant ontologies in previous section, we have detected essential sub-domains to properly model everyday human activities at different levels. These evaluation criteria are available in Tables 3.4.3 and 3.4.3. When a subdomain is modelled in any way in the ontology, X is marked. However, if that domain is highlighted or specially treated in the ontology, this is marked with an increasing number of X (XX, XXX).

When considering human activity representation, basic and obvious variables such as *user, role, location, environment, time, context sources* and

Subdomains addressed by available context ontologies (Ontology Set I)					
Ontology/ Subdomain	CC/PP [126]	CoBrA-Ont [44]	CoDAMoS [166]	Delivery Context [40]	SOUPA [46]
Device	X	X	X	X	
Environment		X	X	X	
Interface				XX (I. Restrictions)	
Location		XXX	X	X	X
Network	X			XX	
Provider					
Role		X	X		
Service			X	X	
Context Source					
Time		X	X		XX
User	X	X	X		X
Imprecision/ uncertainty management					
Message					
Behaviour Granularity		Action	X (Task, Activity)		
Behaviour Model					
Social Interaction					
Implementation available	X	X	X		X
Other Specific Domains Modelled	Capabilities, Profiles	Privacy, agents, compound places, Easy-Meeting (eBiquity Group Meeting)	Platform (HW, OS), hardware resources and SW, service (profile, model, grounding)	Hardware and software (APIs, bluetooth), physical quantities, unit conversions	Policy (also in privacy), EasyMeeting

Table 3.1: Subdomains (Part I) addressed by available context ontologies

Subdomains addressed by available context ontologies (Ontology Set II)					
Ontology/ Subdomain	mIO! [165]	PalSPOT [179]	CONON [222]	PiVOn [99]	Situation Ontology[225]
Device	X	X	X	XX	X
Environment	X	X	X	X	X
Interface	X				
Location	X	X	X	XX	X
Network	X		X		
Provider	X			X (device service provider)	
Role	X			X	
Service	X		X	XX	
Context Source	X	X		X (in device taxonomy)	X (context value domain)
Time	X	X	X	X	XX
User	X	X	X	X	X
Imprecision/ uncertainty management					
Message				X (schedule reminder)	
Behaviour Granularity		X	X	X subtasks	X (boolean & data context operator, atomic/composite situation)
Behaviour Model					
Social Interaction		X		X (companion)	
Implementation available	X	X			

Table 3.2: Subdomains (Part II) addressed by available context ontologies

<table>
<tr><th colspan="7">Subdomains addressed by available context ontologies (Ontology Set III)</th></tr>
<tr><th>Ontology/ Subdomain</th><th>mIO! [165]</th><th>PalSPOT [179]</th><th>CONON [222]</th><th>PiVOn [99]</th><th>Situation Ontology[225]</th></tr>
<tr><td><b>Other Specific Domains Modelled</b></td><td>Games (Paddle) as service</td><td>Indoor/ Outdoor communication routes, vehicles, travelling, device/artefact usage, Snapshot ontology for ADLs</td><td>Home &amp; Office domain, scheduled/deduced activities, furniture status</td><td>Spaces, Visualization Services, Agenda, Schedule, Reminder, Task progress</td><td>Context Value Domain, time period & operators, boolean context operator, atomic /composite situation, academic conference</td></tr>
</table>

Table 3.3: Subdomains (Part III) addressed by available context ontologies

proper *behaviour granularity* levels are necessary. Indoor and outdoor spaces need to be taken into account within the environment. The *role* of a user determines one facet of his/her objectives, which means he/she can have different roles at different times of the day. This is another reason to consider *time* as an essential entity, e.g., each event can have a unique timestamp associated to it. Also, the origin of the *context* information *source* can be crucial to determine its origin or further inferences. Since activity recognition is incremental, some kind of basic atomic *actions*, as well as more generic *activities* and/or *behaviours* should be specified. It is out of atomic events, happening in a certain order, that a given activity can be specified and, therefore, recognized. It is also in this way how higher level context can be inferred out of single sensor events, and so on.

Regarding the supply of assistance, as well as the interaction with the environment, taxonomies on *services*, *devices, interfaces, networks* and *providers* must be taken into account, not only for modelling how to provide users help or support after a human activity has been recognized, but also for proper service grounding and context-aware user adaptation. In this domain, the *mIO!* ontology is the best candidate to completely support *service, network, device*, and *interface* supply.

Another element required to model human activity, is *social interaction*, including *message* exchange among people, but also messages from the system to the user, and vice versa. Also, the *availability* of the ontology is important, so as its maintenance, scalability, etc. The more available and visible an ontology is, the more testing and usage will follow, encouraging also its evolution and adaptation to more concrete and real life domains. In our evaluation, extra features modelled in the analysed ontologies are also highlighted, e.g., OS features, hardware/software platforms, and other more specific domains.

We can observe that features such as modelling *uncertainty, imprecision, and vagueness*, typical of everyday life (and everyday language), are missing in all human context ontologies in Table 3.4.3 and 3.4.3. Formalization of messages as a way of interaction, as well as getting feedback from the user, need to be modelled, as they are samples of social networking or common behaviour. In the next section, we will discuss the studied ontologies' response to each mentioned factor when modelling human behaviour. Common specializations, as well as flaws of the existent ontologies, are to be discussed.

There are very heterogeneous methods for analysing human activity. They specialize on a wide range of scenarios to be tracked, set of activities monitored, used methodology and algorithms, and further specific features such as interaction with other users or scalability of the method. These scenarios are usually surveillance, monitoring of public facilities (train stations, subways, airports), UAV[18] surveillance, patient monitoring, or smart homes.

The sources of context information are varied and the evaluation of activity recognition systems is complex because there is no standard taxonomy of human activities providing, e.g., confusion matrices for each classifier's activities and their respective precision and recall. The way in which the methods execute the data collection and labelling processes is also relevant when it comes to the assessment of different approaches [39]. Additionally, it is not easy to model every context category precisely and generally enough. This is due to the reality that ontology-based activity recognition has also some drawbacks, e.g., it requires good knowledge engineering skills to model the domain. Besides, expressiveness limitations are found in OWL DL, mainly related with the lack of support for temporal reasoning. Directly with OWL, it is not possible to perform interval-based (i.e. overlapping) temporal reasoning, which is crucial for capturing complex temporal relations between activities. Even if ontology-based reasoning has a set of added advantages (discussed in Section 3.2.1 and [51]), ontological reasoning can be computationally expensive [180]. Thus, the design of an appropriate and complete ontology is crucial for ontology based human activity modelling. As an example, we can mention that different data-driven approaches excel at diverse concrete aims, e.g., dealing with missing sensor readings. However, this works well when task models are small and manageable; otherwise,

[18] Unmanned aerial vehicles

an ontology approach seems more efficient [153]. This is just an example that shows that the size of the problem, specific domain, and concrete task are decisive issues to consider when selecting an ontology.

Some kind of different levels in action, activity, or behaviour granularity can be seen only in CoDAMoS, PalSPOT, CONON, PiVOn, and Situation Ontology. While ontologies like CoBrA-Ont only consider atomic actions, CoDAMoS includes a distinction between tasks and activities. PalSPOT also considers two levels of events with actions (e.g. moving arm) and activities (travelling, walking, bathing). However, activity granularity in the CONON ontology occurs only at a unique activity level, differentiating among scheduled and deduced activities. At last, PiVOn includes a recursive subtask structure, while Situation Ontology allows a different approach of granularity based on Boolean and data context operators as well as atomic and composite situations. We could argue that PiVOn's approach is perhaps one of the most flexible solutions to accommodate an infinite number of behaviour granularity levels. However, having a hierarchical categorization of activities helps modularizing and inferring different abstraction levels. This is due to the fact that, as we mentioned earlier, it is important to know the tasks happening, but also the intention, behaviour, or meaning associated to these events.

Looking at the human-computer interaction side, devices, interfaces, network, and services are represented in a large amount of ontologies. However, modelling *messages* in human interaction is only seen in a reduced form, with agenda schedule reminders, in PiVOn. There are no clear ways of modelling the communication back of the system to the user and vice versa. I.e., the kind of interfaces that should be used in each moment and the time to communicate with the user are aspects that should also be context-aware. Social interaction, i.e., human-human, is only modelled in PalSPOT, and consists in a varied taxonomic distinction among acknowledgement, asking and giving opinion, comment, proposal, or request for information. For this matter, PiVOn looks at the social interaction aspect by considering the companion of a user while executing a given task.

If we were to highlight what each analysed ontology stands out for, we could say that mIO! stresses its modelling on device interaction, PalSPOT, on the user activity in the environment and with others, CONON, on activity planning and services, PiVOn, on location and device-based services, and Situation Ontology, in temporal context operations. CC/PP enhances device and network capabilities, while CoBrA masters at locations. CoDAMoS brings out roles and (hardware and software) services, Delivery Context treats (hardware and software) interfaces and networks, and SOUPA, time, locations, and policies.

A summary of the human behaviour ontology review, and its respective coverage in subdomain modelling, can be seen with an estimated evaluation score in Figure 3.4. The ranking is constructed in a straightforward way, assuming each evaluation criteria in Tables 3.4.3 and 3.4.3 as equally important. The histogram, based on a naïve ranking scale, gives an idea of the number of (pre-selected) subdomains modelled by each context ontology. Similarly, the score shows a degree of specialization in a given subdomain. Eighteen categories were used for ontology evaluation (Tables 3.4.3 and 3.4.3), and each ontology gets extra points (one per extra X) if a given domain is specially represented or remarked. The overall grading is normalized between 0-10, according on how many subdomains are satisfied by the ontology and in what strength.

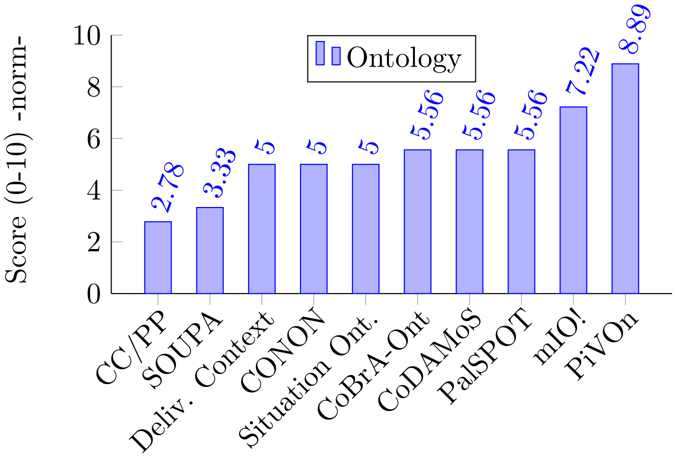


Figure 3.4: Overall ranking for the most representative human activity ontologies for the proposed evaluation criteria.

We are aware that a ranking number does not capture the many dimensions of an ontology, how it can be improved, or the problems it has [214]. Thus, ontology evaluation technologies rather meet their goals by pointing out if an ontology is bad, and in what way, instead of telling how good it is. From the structural point of view, the ontologies could be evaluated as regards their number of pitfalls, i.e., the number of features that could give problems in ontology-based reasoning. Table 3.4 shows the ontologies, their size in triples, and the recommended pitfalls cases to fix, according to OOPS! (OntOlogy Pitfall Scanner) [164]. Due to spatial constraints, we refer the reader to the pitfall catalogue[19] for an accurate key pitfall description and suggested solution. Some of the most commonly identified pitfalls are about creating unconnected ontology elements (*P4*), missing annotations (*P8*), missing domain or range in properties (*P11*), using different naming

[19] OOPS! Pitfall Scanner Catalogue: `http://oeg-lia3.dia.fi.upm.es/oops/catalogue.jsp`

criteria in the ontology (*P22*), or recursive definitions (*P24*). We define the Pitfall Rate evaluation parameter as:

$$\frac{\sum_{i=1}^{n} \#P_i}{\#T} \tag{3.1}$$

where $\#P_i$ is the number of pitfall cases occurring for pitfall type P_i and $\#T$ is the ontology size in number of triples. In this way, the pitfall rate symbolizes the average number of pitfalls per triple. A higher pitfall rate implies the appearance of a larger number of anomalies or errors in the ontology.

Ontology	*# Triples*	*# Classes*	*Pitfalls*	*Pitfall Rate*
CoDAMoS	1291	106	143 *P8* — *P10* — 4 *P11* — 33 *P13* — *P22*	0.141
CoBrA	4144	88	5 *P4* — 186 *P8* — 22 *P11* — 3 *P12* — 67 *P13* — *P22* — 4 *P24* — 5 *Suggestions*	0.071
PalSPOT locont-2.0	5302	199	*P4* — *P5* — 251 *P8* — *P11* — 22 *P13* — 2 *P19* — *P22* — 10 *Suggestions*	0.055
SOUPA _policy	1304	30	12 *P8* — 3 *P11* — 7 *P13* — 1 *Warning*	0.025
Delivery-ContextAll	22573	134	*P4* — 37 *P11* — 97 *P13* — *P22* — *P24* — 4 *Sugg*	0.006
CCPP schema-20030226	19	134	Free of bad practice detectable by OOPS! Pitfall Scanner	0

Table 3.4: Ontology pitfall evaluation

We are also aware that defining ontology quality is difficult, since it depends on different approaches. As a result, we can also appreciate that the semantic evaluation ranking in Figure 3.4 is independent of the structural assessment in Table 3.4. The majority of methods for ontology evaluation concentrate on structural evaluations, in a similar manner to what validators do. This is because semantic evaluation is subjective and application and domain-dependent. Other methodologies that can help choosing the right ontology include ONTOMETRIC [140] and an adaptation of the AHP method[20], to help knowledge engineers choose the appropriate ontology for a given domain. Identified characteristics for the evaluation include tools, language, content, methodology, and costs [140]. Other evaluation methods can be found in [194] or COAT[21] - Cognitive Ontology AssessmenT, but most of the methods do not provide a versatile software interface such as OOPS'.

Following a methodology to guide ontology development has proven to be useful [165], just as reusing knowledge resources and attending to good practices in the ontology development. However, when searching for an existing ontology for a given domain, selecting and reusing context ontologies can become difficult due to the different purposes and requirements for which the ontologies are designed. For instance, in our domain, a hierarchical classification (of human actions and activities) for a granular behaviour disclosure is required to incrementally infer new information from a collection of temporally evolving and atomic context data. However, specific requirements are not always, nor often, supported and each application scenario will normally impose different requirements.

Elements such as location and time are essential when managing historical context data, in order to provide intelligent learning algorithms that can offer services after recognizing an activity. We believe that the possibility of associating machine learning behavioural models to each behaviour in the ontology can provide modular and proactive capabilities without depending on specific implementations of context-aware systems, but rather having a formulation in the ontological model itself. Some approaches close to this paradigm are hybrid systems such as *COSAR* context aggregation and reasoning middleware and its ontological/statistical hybrid reasoners. Their *Derivation of Possible Activities* algorithm, executed by an off-line ontological reasoning module, takes an empty ABox and TBox as input to output a matrix of symbolic locations and their correspondent inferred human activities [178].

It becomes evident that human activity ontologies need to further consider the modelling of imprecise and uncertain information for more accurate representation of everyday human tasks and human language. This is an aspect that no ontology tackles at the moment. Other issue to deal with is modelling social interaction, both virtual (for human-to-computer messages) and physical (for human-to-human messages). As we mentioned, having an adequate level of action granularity (e.g., actions, tasks, activities, behaviours, etc.) is crucial for a specialized and incremental discovery. Besides, a standardized common representation of universal entities, such as time, geographical indoor and outdoor locations, as well as environmental

[20] Analytical Hierarchy Process, a measurement method based on preferential ordering

[21] COAT (Cognitive Ontology AssessmenT) tool: https://code.google.com/p/ontoeval

conditions would greatly help in these processes. At the same time, relations such as ownership of objects, rights, services, or privacy and service access are common issues of interest, usually modelled in different ways. Unfortunately, this causes the reuse and mapping of several heterogeneous ontologies to require long time spent in curating and matching ontology concepts.

When considering the real expressive power and usability of the reviewed ontologies, it is also important to note that OWL 2 language is powerful for expressing knowledge, context information, as well as relations among entities. However, OWL 2 is insufficient to model context relations and rules with the form of cyclic relations [179] (e.g. relations such as *isColleagueWith* in the rule in Section 3.4.2, Example 2). Therefore, the ontologies discussed require of an integration with a rule language (such as e.g., SWRL [swr] or SPIN[22][spi]) in order to express more complex and real life context rules. The combination of DL with rule-based systems improves the reasoning capabilities. Rule-based languages enable definition of consistency rules reducing ambiguity in the context information and thus maintaining and improving the information quality [99]. For instance, SWRL (used e.g., in PiVOn ontology [99]) extends the semantics of OWL and defines antecedent-consequent rules and built-in (math, comparisons, string and time) operators. Another example of rule-based inference support over OWL graphs is Jena Semantic Web Toolkit[23], used, for instance, in CONON ontology [222]. Other example is Flora-2 object-oriented knowledge base language[24], the inference engine in CoBrA ontology. It can be concluded that the final expressivity of the ontology-based application will be a result of the combination of the elected ontology and its coupling with the chosen rule language, as an extension of OWL axioms, to express context rules.

We can clearly point out that the integration of different methodologies, i.e. data-driven and knowledge-based ones, could help overcome current limitations in scenarios with several actors, providing semantics to social activities, user identification according to behaviour semantics, etc. Current hybrid approaches such as [82, 178] have shown that these types of combinations can enhance the response of data-driven approaches as the environment complexity and the context-awareness needs increase. In addition, knowledge-based approaches also could take advantage of features such as noise tolerance and uncertainty handling, inherent to most used data-driven activity recognition models. Following chapters will explore these ideas.

What can be appreciated from this survey is that most of the works require a data-intensive-driven first approach to robustly identify the most basic level actions or activities. Based on these set of actions and activities,

[22] SPIN Rules: `http://spinrdf.org/`

[23] Jena Semantic Web toolkit: `http://jena.apache.org/`

[24] FLORA-2: `http://flora.sourceforge.net/`

recognized first through precisely specified and robust models, further context assumptions can be integrated into the models to better handle the uncertainty inherent to the environment context. As in ubiquitous computing applications it is not possible to specify all possible cases to recognize human behaviours, the implicit reasoning capabilities of knowledge-driven methods allow for more flexible and context-aware models, i.e., more knowledge can be expressed without explicit specification nor knowledge redundancy [225]. The latest research shows the benefits of introducing hybrid approaches to take advantage of each technique's best strengths. Combining ontology-based context reasoning together with data-driven algorithms has shown to be a promising path to be explored. An example is combining ontological reasoning with computer vision research [82]. Other works following these lines, with different (statistical, ontological) reasoning modules [178, 179], show that the usage of hybrid approaches with a preliminary phase on data intensive methods can ease the way later, when inferring higher level activities through knowledge-driven approaches. As time is not a feature inherently treated in knowledge-driven approaches such as logic-based systems, having hybrid methods with a first data-driven preprocessing stage appears to be the right direction to benefit from both data- and knowledge-driven computing paradigms. As ontological reasoning can be computationally expensive, this type of combinations would achieve the best performance and efficiency from (time-dependent) data-driven methods, and obtain the best adaptation for context-awareness in each case.

In this chapter, an evaluation taxonomy for learning procedures, methods, models, and modelling capabilities was proposed. We also presented a set of upper ontologies designed to represent human activity, as well as domain ontologies that can serve the same aim in context-aware intelligent environments. A complete set of evaluation criteria was introduced to assess the current ontologies, having as main focus the different subdomains required for human behaviour representation, learning, and inference. The evaluation was performed by analysing different useful domains and was concluded by giving an overall score to each ontology. Furthermore, as the semantic quality of the ontology ultimately depends on the specific domain to be modelled among other multiple aspects, we analysed the structural problems, or *pitfalls*, found in each ontology. We can confirm that the broader an ontology is, the more situations will be possible to be modelled, in order to assist the users in their daily activities, and the less usable the ontology will be in order to achieve a particular goal [165]. However, the more specific the ontology is, the fewer possibilities exist for reuse, but the more usable the ontology is. Next chapter will propose a fuzzy ontology that can serve our requirements and will model the missing aspects found in the described existing solutions in this chapter.

Chapter 4

Methodology proposal: A fuzzy ontology for human behaviour modelling and recognition

> Travel is fatal to prejudice,
> bigotry, and narrow-mindedness
>
> Mark Twain

In the previous chapter we analysed the state-of-the-art on existing ontologies for human activity recognition and identified missing elements regarding subdomains or types of information that they can represent and handle [71]. Therefore, in this chapter, we propose a novel ontology that can provide a solution for the missing elements identified in the literature. We first describe the design principles used to represent human behaviour in a crisp subset/core of the ontology and then, we detail the fuzzification procedure to obtain the proposed fuzzy approach.

Among the different knowledge engineering methodologies, we mostly followed *NeOn* Ontology engineering methodology [206] to implement our approach and all use cases ontologies developed in this chapter. Some highlighted aspects of this methodology are the ontology resource reuse, requirements specification, development of required scenarios, and dynamic ontology evolution. NeOn targets software developers and ontology practitioners, it has dynamic guidelines for ontology evolution and treats context dimension and distributed collaboration. As human activity recognition requires constant update of input sensor data and the ability to adapt to changes in behaviour, NeOn allows the evolution of the ontology.

4.1 Ontological modelling of human actions, activities and behaviours in Ambient Intelligence

First we modelled a basic (crisp) ontology that gathers concepts from different basic subdomains. We can distinguish among four main entities: Users, Environment, Activities, and the corresponding interactions among them (Relationships). The three first elements can be considered the core of the ontology, and all together define the *what, who, when*, and *where* for each relevant event to be annotated:

1. *Users.* Users can be divided into two categories: A *Single User* is used to represent a unique user performing activities. Aspects to consider when learning behaviours contain the user's location, role/position, calendar, the user social network, etc. On the other hand, the second category *Multi-user/Generic User* involves a group of users sharing a common behaviour or objective (meeting attendants, visitors, students, etc.). Since this kind of behaviour embraces a group of people performing the same action, they are not considered individually. We define a Generic User class as a way to represent an abstraction of a group of users. The *Generic User* shares some properties with *Single User* actions and activities, such as roles or access rights, etc. This class is also useful when personal data about users is not known, but is relevant as an observed entity in the environment.

2. *Environment.* It is an organized hierarchy of locations, models, and generic and specific features of each kind of space. For example, in the office domain, environments are offices, meeting and lecture rooms, auditoriums, kitchen, toilets, etc. Different levels to track indoors positioning may include floor and room numbers, while outdoor locations may refer to open spaces or means of transport. A location can have associated measurements such as *Humidity, Temperature, Lighting, NoiseLevel*, or *Pressure.* For a finer grained environment, objects (e.g. doors, curtains, windows) can have, e.g., an *Aperture* state and rooms and locations a (seating) *Capacity.*

3. *Actions, Activities, and Behaviours.* We distinguish among three types of events or activity granularity levels.

   - *Actions* or atomic events with a *timestamp*, e.g.: *OpenDoor, MoveObject, TurnLightOff, WalkBy, BeObservedInLocation*, etc. This is the lowest granularity degree of representation.
   - *Activities*, considered as single actions with an inherent "purpose", or composed of a set of different actions. An activity represents an intermediate granularity level of representation and has

a *startDatetime* and *endDatetime*. E.g.: *TakeCoffee, AttendConference, GroupMeeting, VideoCall, SendEmail*, etc. An activity is defined by a set of compulsory actions plus a set of optional actions, where some of them can have temporal execution interdependencies, e.g. *MakeCoffee, SpeakAtConference, VisitLocation*, etc.

- *Behaviours*, a sequence of activities and/or actions. A behaviour is defined by a set of compulsory actions or activities plus a set of optional actions or activities, where some of them can have temporal execution interdependencies. e.g., the behaviour *CoffeeBreak* includes the Action *ExitOffice*, the Activity *MakeCoffee* or *TakeCoffee*, and the Action *EnterOffice* in this order. The difference between activity and behaviour is that an activity is always the same regardless the context, while a behaviour is always defined and valid within a concrete context. For instance the activity *Running* is always defined in the same way. However, if *Running* has a specific meaning or goal within a context, then it can be (part of) a specific behaviour.

Any user can perform (atomic) actions, activities and/or behaviours. These are structured in a hierarchy of ascending abstraction. Depending on the actors in the environment, they will be individual or collective (social) activities. It is important to mention the possibility of *recursion* in the definition of activities and behaviours: both can be defined semantically (through the *involves* property) as a set of actions and (sub) activities. For instance, the activity *going to bed* is composed of the activities *having infusion* and *taking pills*, and the actions *putting pyjama* and *brushing teeth*. This design allows us to work in different degrees of granularity and to decompose complex activity and behaviour recognition stages into simpler procedures.

4. *Relationships*. Object and data properties model interaction among users and the environment that can serve to specify behaviours. Relationships can link *Single User*s, *Generic User/Multi-user*s, or users and environment elements.

An excerpt of types of users, locations, and a subset of Actions and Activities, as well as object and data properties can be seen in Figures 4.1, 4.2, 4.3, and 4.4. In addition, Figure 4.5 represents a systematic relationship among the activity, user, and environment context. Since the full proposal cannot be shown due to space limitations, we make the complete ontology available through web[1][HAR]. Our approach is focused on the office domain

[1] Fuzzy Human Behaviour Ontology and experiments: `http://users.abo.fi/ndiaz/public/FuzzyHumanBehaviourOntology/`

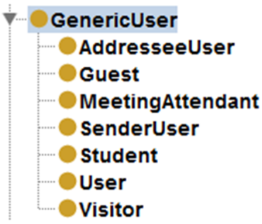


Figure 4.1: Excerpt of *User* subclasses in the ontology (partial).

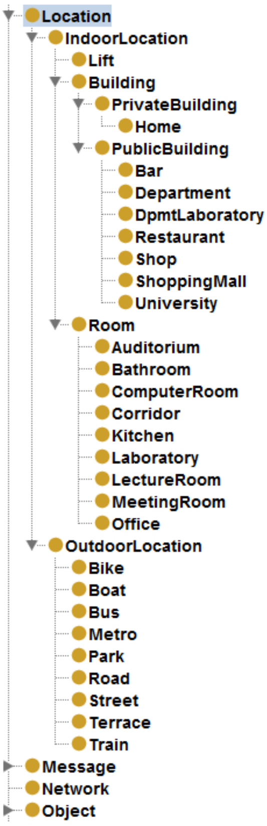
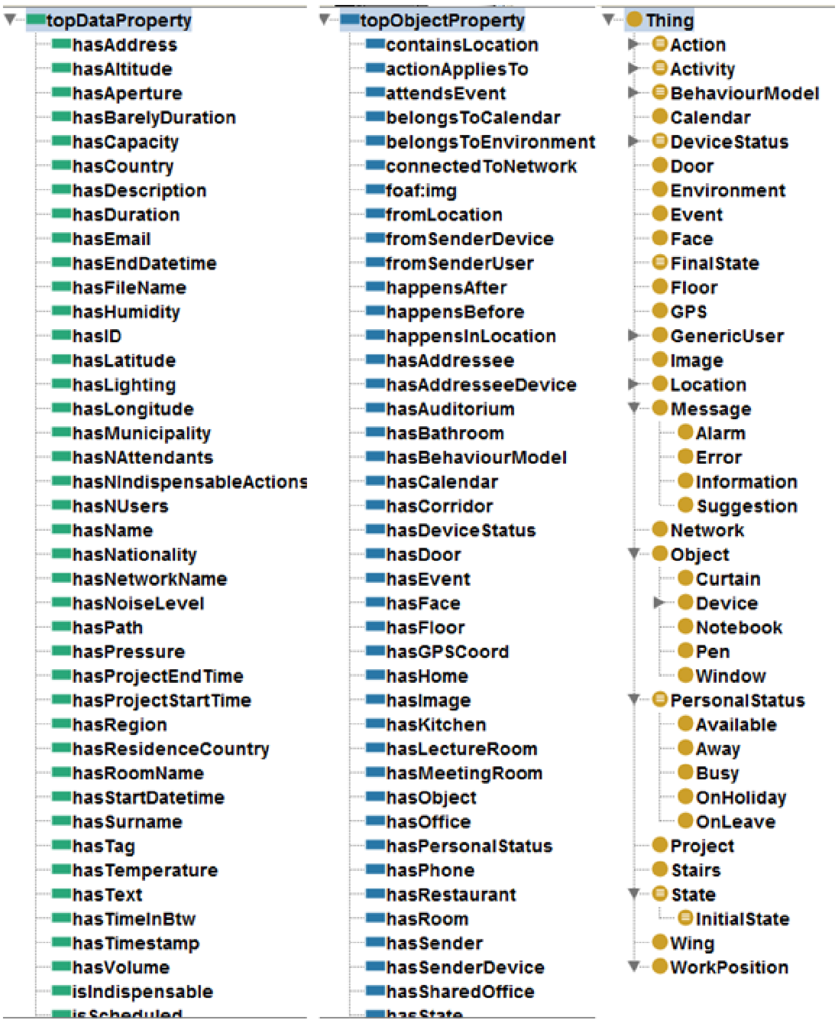


Figure 4.2: Excerpt of *Location* subclasses in the ontology (partial).



- topDataProperty
  - hasAddress
  - hasAltitude
  - hasAperture
  - hasBarelyDuration
  - hasCapacity
  - hasCountry
  - hasDescription
  - hasDuration
  - hasEmail
  - hasEndDatetime
  - hasFileName
  - hasHumidity
  - hasID
  - hasLatitude
  - hasLighting
  - hasLongitude
  - hasMunicipality
  - hasNAttendants
  - hasNIndispensableActions
  - hasNUsers
  - hasName
  - hasNationality
  - hasNetworkName
  - hasNoiseLevel
  - hasPath
  - hasPressure
  - hasProjectEndTime
  - hasProjectStartTime
  - hasRegion
  - hasResidenceCountry
  - hasRoomName
  - hasStartDatetime
  - hasSurname
  - hasTag
  - hasTemperature
  - hasText
  - hasTimeInBtw
  - hasTimestamp
  - hasVolume
  - isIndispensable
  - isScheduled

- topObjectProperty
  - containsLocation
  - actionAppliesTo
  - attendsEvent
  - belongsToCalendar
  - belongsToEnvironment
  - connectedToNetwork
  - foaf:img
  - fromLocation
  - fromSenderDevice
  - fromSenderUser
  - happensAfter
  - happensBefore
  - happensInLocation
  - hasAddressee
  - hasAddresseeDevice
  - hasAuditorium
  - hasBathroom
  - hasBehaviourModel
  - hasCalendar
  - hasCorridor
  - hasDeviceStatus
  - hasDoor
  - hasEvent
  - hasFace
  - hasFloor
  - hasGPSCoord
  - hasHome
  - hasImage
  - hasKitchen
  - hasLectureRoom
  - hasMeetingRoom
  - hasObject
  - hasOffice
  - hasPersonalStatus
  - hasPhone
  - hasRestaurant
  - hasRoom
  - hasSender
  - hasSenderDevice
  - hasSharedOffice
  - hasState

- Thing
  - Action
  - Activity
  - BehaviourModel
  - Calendar
  - DeviceStatus
  - Door
  - Environment
  - Event
  - Face
  - FinalState
  - Floor
  - GPS
  - GenericUser
  - Image
  - Location
  - Message
    - Alarm
    - Error
    - Information
    - Suggestion
  - Network
  - Object
    - Curtain
    - Device
    - Notebook
    - Pen
    - Window
  - PersonalStatus
    - Available
    - Away
    - Busy
    - OnHoliday
    - OnLeave
  - Project
  - Stairs
  - State
    - InitialState
  - Wing
  - WorkPosition

Figure 4.3: Excerpt of Classes, Data and Object Properties in the ontology (partial).

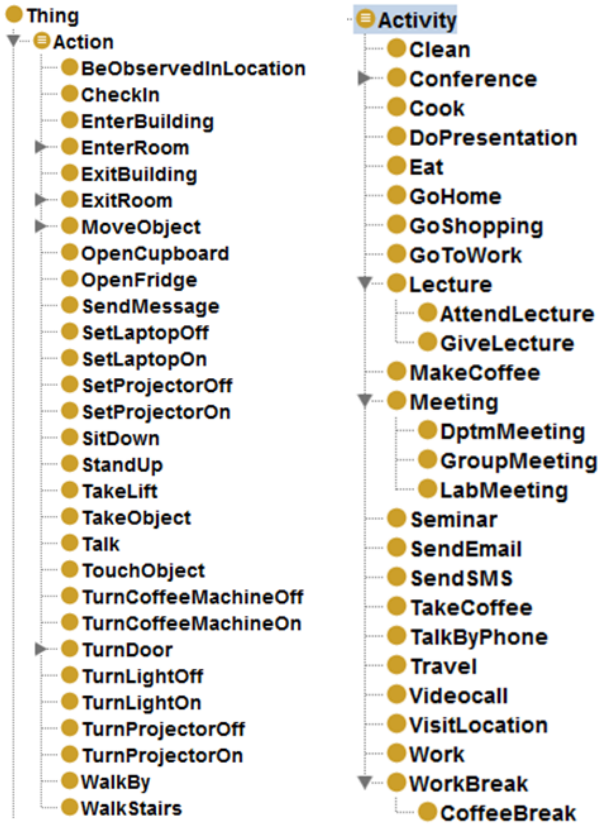


Figure 4.4: Excerpt of *Action* and *Activity* classes in the ontology (partial).

and reuses some hierarchy concepts from existing ontologies, e.g., the indoor/outdoor hierarchy from the *CONON* ontology [222] or environment features from *CoDAMos* ontology [166]. We also include other features of interest that we considered relevant in a human activity ontology:

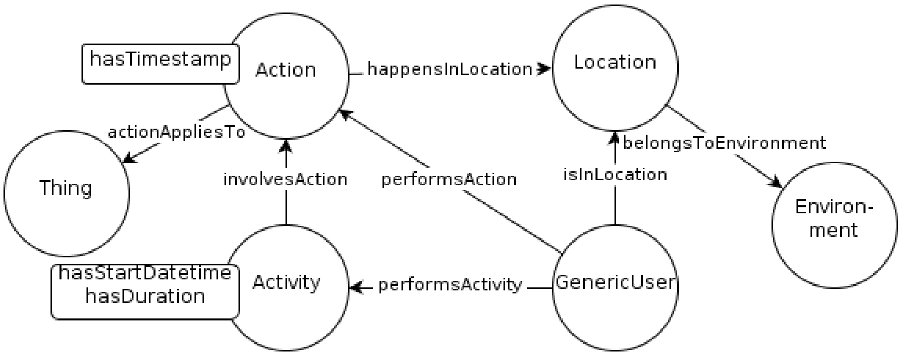


Figure 4.5: Main relationship among *User, Activity*, and *Environment* in the modelled context ontology.

- *Activity duration and concurrency*: Several activities can be performed, by the same or different people, at the same or overlapping times. Also, some events resulting from user actions, remain done for another user later, e.g., turn the lights on. The next users may not have to perform that action in order to complete the same activity as the first user. To model these situations, the Boolean data property *remainDone* of the class *Action* indicates when an action required by an activity can remain done or "active" to someone else.

- *Activity characterization and indispensable actions*: Not all users carry out a given behaviour or activity on the same way. Therefore, an activity can be performed according to more than one behaviour model, one per user. In order to know, in an efficient manner, if certain action is absolutely required to perform a given activity, the object property *isIndispensableForActivity (Action, Activity)* can be set to avoid further computations. The property *isIndispensableForBehaviour (Action OR Activity, Behaviour)* works in the same way for behaviours.

- *Messages or alerts*: Alerts are useful as reminders of forgotten actions or to warn about potentially hazardous situations. Types of messages are *Error, Alarm, Information*, or *Suggestion*; modelling the message recipients happens through *Sender* and *Addressee Device*, and *User* classes, respectively.

4.2 Case study 1: Ontological modelling of movement and interaction with a 3D depth sensor ontology

Microsoft Kinect has attracted great attention from research communities, resulting in numerous interaction and entertainment applications. Including automated semantic reasoning in these settings would open the doors for new research, making possible not only to track but also understand what the user is doing. We developed a 3D depth sensor ontology, modelling different features regarding user movement and object interaction (through voice and movement), because we believe in the potential of integrating semantics into data-driven approaches such as computer vision. As 3D depth sensors and ontology-based applications improve, the ontology can be used, for instance, for activity recognition, together with semantic maps for supporting visually impaired people or in assistance technologies, such as remote rehabilitation.

Semantic modelling of human movement and interaction could greatly benefit existing data-driven (e.g., computer vision) approaches, increasing context-awareness and potentially, activity recognition rates [68]. Using vision based techniques has substantial disadvantages, as most of them store the images, and become intrusive and privacy compromising. Since 3D depth sensors do not store the image itself, but a skeleton structure, they add an advantage towards traditional data-driven approaches [122] (HMM, SVM, etc.)

4.2.1 Semantic approaches for computer vision and depth data

Due to Kinect multimodal features such as gesture and spoken commands, different *UbiComp* applications have been recently developed. For instance, the combination of Kinect with an airborne robot [97] to enable automatic 3D modelling and mapping of indoor environments. An interesting initiative in this area is Kinect@Home[2][125], a crowd-sourcing project for large 3D datasets of real environments to help robotics and computer vision researchers, through vast amounts of images, to improve their algorithms. Another project, *Kinect Fusion* [111], allows for real-time 3D reconstruction and interaction using point-based 3D depth sensor data. An application example is touch input enabled arbitrary surfaces.

The following example illustrates with OWL 2 axioms the activity *Take Medication*, that can serve to monitor an elder:
$NataliaTakingMedication \equiv Activity \sqcap \exists isPerformedBy.(Natalia \sqcap \exists per-formsAction.(OpenPillCupboard \sqcap (TakeObject \sqcap \exists actionAppliesTo.Natalias-$

[2] Kinect@Home `http://www.kinectathome.com/`

$Medication) \sqcap (TakeObject \sqcap \exists actionAppliesTo.Glass) \sqcap FillGlassWithWater \sqcap Drink))$.

In [80] ontology-based annotation of images and semantic maps are realized within a framework for semantic spatial information processing. An XML description language for describing the physical realization of behaviours (speech and behaviour) is the Behaviour Markup Language (BML)[3], which allows representation of postures and gestures for controlling verbal and non-verbal behaviour of (humanoid) embodied conversational agents (ECAs). However, to the best of our knowledge, there is no current solution integrating the performance power of computer vision technologies, together with a formal semantic representation of the user, its movement and interaction with the environment, to achieve automatic knowledge reasoning. In next section we propose an ontology for combining data-driven and knowledge-based paradigms.

4.2.2 An exercise-workout 3D depth ontology for Ambient Intelligence, remote monitoring and tele-rehabilitation

As there does not exist any automated semantic reasoning for modelling movement and interaction within computer vision technologies and 3D depth sensors, we propose an ontology to distinguish among human movement, human-object interaction and human-computer interaction. The Kinect ontology[4][Kin] aims at representing 3D depth sensor information generally, but at this stage it is based upon two main Kinect modules. The first and most basic one is Kinect Core, and represents the Natural User Interface (NUI), which is the core of the Kinect for Windows API, and represents the most relevant concepts from Kinect Interaction and Kinect Fusion APIs [Kinect for Windows]. The second module of the ontology consists of practical extensions for modelling and recognizing human activity. Some of the classes represented are the *Kinect Sensor*, *3D Model* with the user' skeleton or *Kinect 3D Volume* and *Kinect Audio. Kinect Interaction* provides several ways to interact with a Kinect-enabled application. The natural gestures, as a way of touch-free user interactions, allow the sensor to operate in a range of 0.4 to 3-4 m. The types of interaction are modelled with gestures (gripping, releasing, pushing and scrolling) (Fig. 4.6).

This class generates interaction streams which are bound to a control, i.e., an action that allows computer interaction. A *Control* is an action performed when an interaction gesture is recognized. The set of interactive controls are classified on video, images or text. An *Interaction Stream*

[3] BML: `http://www.mindmakers.org/projects/bml-1-0/wiki`

[4] Kinect Ontology: `http://users.abo.fi/rowikstr/KinectOntology/`

represents the supply of interaction frames as they are generated by a *Kinect-Interaction*. Each *InteractionFrame* has a timestamp.

Kinect distinguishes among two types of *Tracking Modes*, *default* or *seated*. Both modes can track 2 out of 6 users, but only one can be active at once.

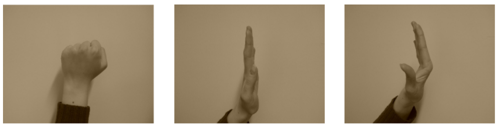


Figure 4.6: Some available interaction gestures: a) Grip b) Release c) Press

By using ontology-based modelling, different kind of users can be defined as follows:

$SeatingUser \equiv User \sqcap (\exists\, isTracked.True) \sqcap (\exists\, hasSeatedTrackingMode - Active.True)$.

$StandingUser \equiv User \sqcap (\exists\, isTracked.True) \sqcap (\exists\, hasDefaultTrackingMode - Active.True)$.

$TrackedUser \equiv User \sqcap (\exists\, isTracked.True) \sqcap ((\exists\, hasDefaulTrackingMode - Active.True \sqcup (\exists\, hasSeatedTrackingModeActive.True))$.

$InteractingUser \equiv User \sqcap (\exists\, isTracked.True) \sqcap (\exists\, hasArm.Arm) \sqcap (\exists\, hasHand.Hand) \sqcap (\exists\, hasInteractionMode.(GrippingInteractionMode \sqcup ReleasingInterac-tionMode \sqcup PressingInteractionMode))$.

Kinect' *Skeleton* class identifies a *User* and is represented with a bone and joint hierarchy, which refers to the ordering of the bones defined by the surrounding joints. Our ontology allows to express relations concerning bones and joints, where the bone rotation is stored in a bone's child joint, e.g., the rotation of the left hip bone is stored in the *HipLeft* joint (See Fig. 4.7, right)[5]. The skeletal tracking includes rotations of each bone joint and orientations of each bone.

The *Hand* class has a set of properties that represent its state, e.g., the user the hand belongs to, whether the hand is primary for that user, whether the hand is interactive, gripping or pressing. *Arms*, in the same way, are provided with an arm state.

Kinect Extensions Ontology

A set of relevant classes is defined next to make sense on body, objects and actions interactions.

[5] Bones are specified by the parent and child joints that enclose the bone and their orientation (x,y,z). For example, the Hip Left bone is enclosed by the Hip Center joint (parent) and the Hip Left joint (child) [Kinect for Windows].

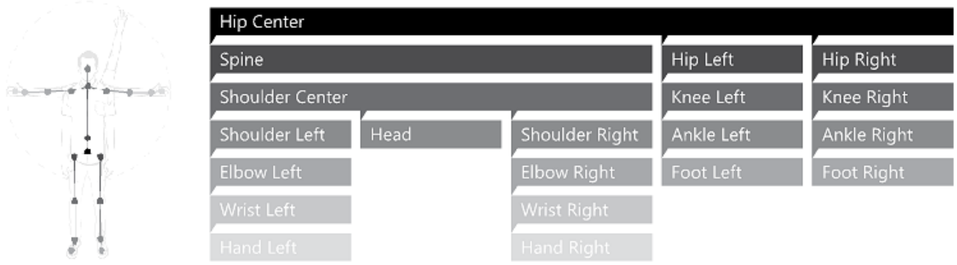


Figure 4.7: Left: Skeleton, bones and joints. Right: Joints hierarchy. [Kinect for Windows]

The class *User* identifies the person behind the Skeleton model. A user is modelled with the correspondent arms (and hands) and a set of properties that, e.g., may identify him as *PrimaryUser*[6].

Body Movement mainly represents actions executed with body limbs and articulations. Different kind of movements include to rotate, bend, extend and elevate. These can have a clockwise direction (e.g. *RotateWristClockwise*), a direction (*ElevateFootFront*), a degree or a body part to which they apply (*LeftBodyPart*).

Any physical *Object* and its properties such as dimensions, (partial) colours or number of voxels can be represented, for instance, to recognize activities such as experiments involving volume measurements. *Object actions* model interaction between objects or among users and objects thanks to Kinect Fusion API module. Examples of interactions between user and objects include to *grab, release, touch, click* etc.

The Spatial Relations Ontology [106] is reused to express physical space relations of objects as well as how they are placed or how they interact with each other, e.g. *contains, disjoint, equal and overlaps.*

The main classes, data and object properties of the Kinect Ontology are presented in Table 4.1.

4.2.3 Ontology-based human activity reasoning with the 3D depth sensor ontology

Figure 4.8 presents the structure of the Exercise & Workout Sub-Ontology, where the goal is to precisely model the specific movements a user performs, e.g., through the exercise duration, repetitions and quality or intensity (*Low, Medium, High*) performed.

[6] Kinect Interaction layer decides which of the tracked users is primary and assigns him an ID and a primary hand, although both hands are tracked [Kinect for Windows]

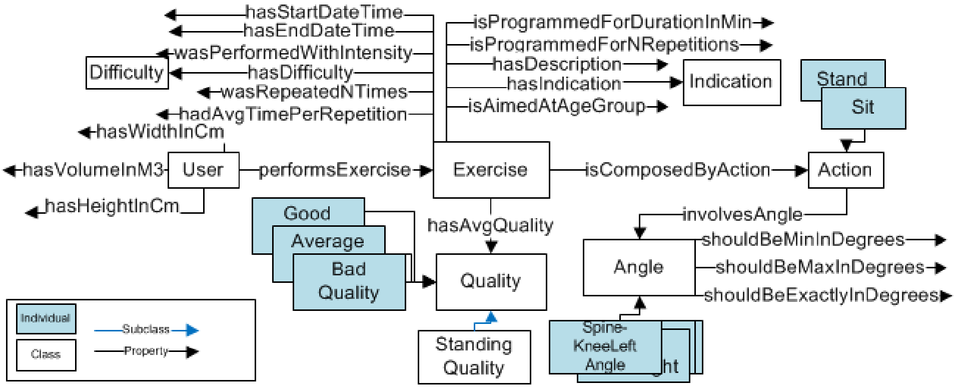


Figure 4.8: Exercise & workout sub-ontology

OWL Classes	**OWL Data Properties and Object Properties**
BodyMovement, BodyPart, ObjectAction, Exercise, Angle, (Image, Text, Video-)Control, Exercise(-Difficulty, Frequency, Intensity, Quality) Grammar, HandState, Indication, Location, Object, Orientation, Kinect-(Audio, Interaction, Sensor), Dictation, SpeechRecognitionEngine, TrackingMode, Bone, BoneJoint	*hasStart/EndDateTime, wasRepeatedNTimes, hadAvgTimePerRepetition, shouldBeMin/Max/ExactlyInDegrees, hasDescription, isProgrammedForNRepetitions, IsProgrammedForDurationInMin, hasCoordinateX/Y/Z, hasHeightInCm hasDifficulty, hasIndication, hasAvgQuality, performsExercise, isComposedByAction, involvesAngle, hasOrientation, hasSourceLocation, interactsWith, detectsKinectAudio, hasLoadedGrammar, hasActiveTrackingMode, detectsInteraction/Object, activatesControl, hasBoneHierarchy, isLocatedIn, hasSpatialRelation, hasInteractionMode, hasArm/-Hand, hasSpeechRecognition, representsUser*

Table 4.1: Kinect Ontology Classes, Data and Object Properties (partial)

In order to model human activities and behaviours, the state of environment variables and body postures can be abstracted so that identifying changes of interest is possible. Since existing statistical methods have demonstrated to be robust in activity monitoring [109], the Kinect ontology is intended to support these by adding context-awareness to the end-user application. For instance, long-term queries could be done, since having semantic knowledge adds the capability of integration with other sensor information, allowing for user-customization of the smart environment. Therefore, we focus on representing simple, higher level actions (lay down, washing hands, etc.) and facilitating the finding of longer term changes. Examples of the ontology in use are:

Example 1: Defining basic movement (*Stand, BendDown, TwistRight, Move-Object*, etc.) can be mapped to OWL 2 DL, e.g., the Action *Sitting*, would be of the form:
$\forall\, U \in User$:
$Sitting \equiv U \sqcap \exists performsAction.(Sit \sqcap hasStartDatetime.dateTime)$.

Example 2: When defining an activity, e.g. *Sit_StandExercise* workout, the amount of series done in a given time as well as the exercise quality can be measured. These values can be predefined according to medical parameters, e.g., the difficulty faced when sitting/standing as well as the stretching of the back when standing can be expressed in DL as:
$\forall\, Si \in Sit, \forall\, St \in Stand, \forall SitStand \in Sit\_StandEx$:
$BadQualitySitStandExercise \equiv \exists performsExercise.(SitStand \sqcap isCompo-sedByAction.(Si \sqcap St) \sqcap involvesAngle.(LowerUpperBackAngle \sqcap < 175$ $hadAngleValue)$.

Example 3: Historic analysis can be provided through measurements performed while doing certain activity, to monitor posture quality. E.g., having the back less straight than a year ago could be notified to make the user aware of his posture habits[7]. In next example, ΔA represents some angle variation in degrees (*float* in OWL 2), and *T*, ΔT, and expressions such as *2h*, represent time or time variations (*xsd:dateTime* literals in OWL 2):
$\forall U \in User, \forall P \in Phone, \forall St \in Stand, \forall UpperBackA1, UpperBackA2 \in Upper-BackAngle, \forall\, A1, A2 \in int, \forall D1, D2 \in datetime$:
$performsAction(U, St) \wedge involvesAngle(St, UpperBackA1) \wedge hasValue(Upper-BackA1, A1) \wedge hasDateTime(UpperBackA1, D1) \wedge involvesAngle(St, UpperBack-A2) \wedge hasValue(UpperBackA2, A2) \wedge hasDateTime(UpperBackA2, D2) \wedge ((A1-A2) > \Delta A) \wedge T2 = (T1 + \Delta T) \wedge hasPhone(U, P) \rightarrow SendSMS(P,$ "Your back is not as extended as a year ago"$)$.

Example 4: An office worker can be notified when he is not having

[7] Note: due to the need of additional operators (+,-), the following examples are expressed in first order logic (FOL)

straight back and neck. The following FOL rules express these situations:
$\forall\, S \in Sit\,, \forall\, NeckA \in NeckUpperBackAngle\,, \forall\, V \in int, \forall P \in Phone$:
$isCurrently(Natalia, S) \wedge isInLocation(Natalia, NataliasOffice) \wedge involvesAngle$ $(S, NeckA) \wedge hadAngleValue(NeckA, V) \wedge V < 175 \wedge hasPhone(Natalia, P) \rightarrow$ $SendDoubleVibrationAlarm(P,$"Bad posture!")
Or when the user has been sitting for too long:
$\forall\, T, CurrentTime \in datetime, \forall\, P \in Person, \forall S \in Sit$:
$executesAction(P, S) \wedge hasEndDateTime(S, T) \wedge ((CurrentTime - T) > 2h) \rightarrow$ $sendAlarm(P,$"Stand up and stretch your legs!").

The integration with other physiological data such as heart rate, sleep quality or stress, from sensors such as accelerometers, can be as well integrated for more complete assessments of every day functions or tasks.

As a result, the developed OWL 2 ontology ($\mathcal{ALC}$ DL expressivity) is composed of 164 classes, 53 object properties, 58 data properties and 93 individuals, and it is based on the Kinect for Windows API, Kinect Natural User Interface, Kinect Interaction, Fusion and Audio modules.

4.2.4 Implementation and integration into a low-power, local-storage architecture for Smart Spaces

The ontology developed was deployed into a prototype system that can be used to monitor remotely rehabilitation exercises. The final interface with skeleton tracking and exercise rehabilitation is in Figure 4.9.

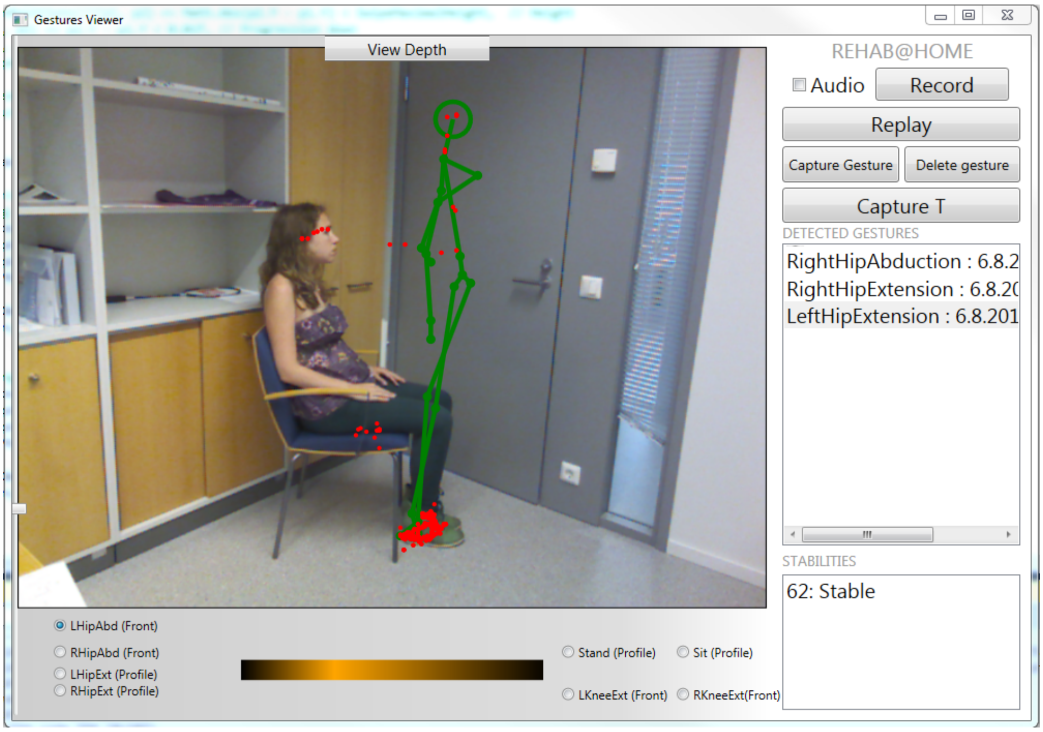
We deployed a semantic RDF store for ontology-based knowledge representation and publish/subscribe-based rules. The store runs on an (low power) Atom board box running M3.

Within the Active Healthy Ageing platform project (AHA)[8] we developed two extra ontologies. An AHA ontology for sensor data interoperability to be integrated into the Personal Health Labs API, and a security and privacy ontology [105], developed to preserve, at triple level, privacy and security of personal critical data. Part of the security ontology for triple level access-control in a Smart Space can be seen in Figure 4.10, and more details are in [105].

Motivation: remote rehabilitation

The motivation to implement a remote rehabilitation application is due to the existence of areas such as Turku surroundings, which is a Finnish archipelago of islands, where many of them are populated with few people. The particular challenge in this case is to provide health related services,

[8] Active Healthy Ageing platform project: `http://www.eitictlabs.eu/innovation-areas/health-and-wellbeing/`



DETECTED GESTURES

RightHipAbduction : 6.8.2
RightHipExtension : 6.8.20
LeftHipExtension : 6.8.201

STABILITIES

62: Stable

● LHipAbd (Front)
○ RHipAbd (Front)
○ LHipExt (Profile)
○ RHipExt (Profile)
○ Stand (Profile)
○ Sit (Profile)
○ LKneeExt (Front)
○ RKneeExt(Front)

Figure 4.9: Activity recognition in tele-rehabilitation [63].

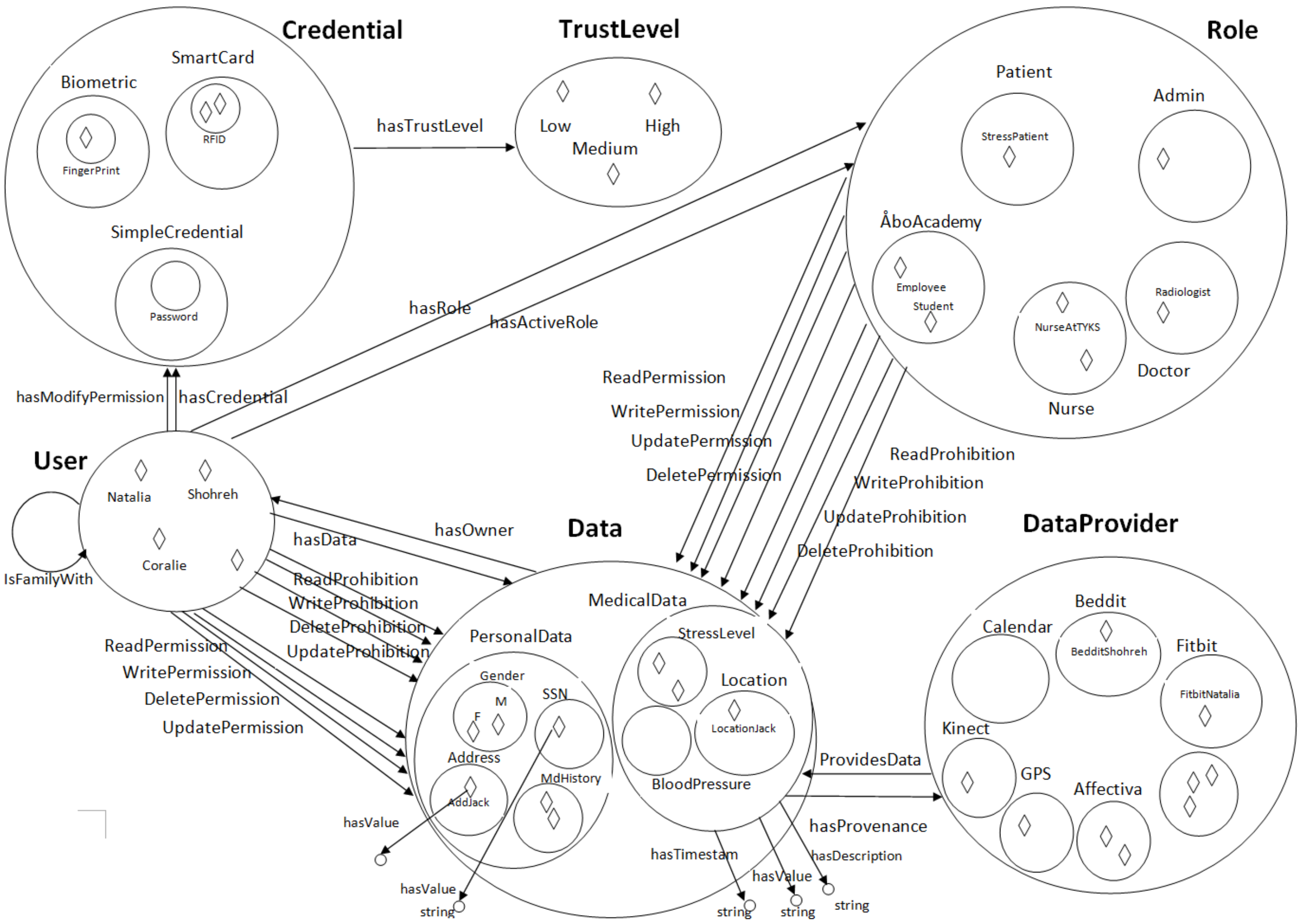


Figure 4.10: Privacy and security ontology for triple-level access control in Smart Spaces [105].

in particular during the autumn and spring, when the weather conditions make it difficult to reach the islands (e.g. when the ice is not strong enough to carry cars, but strong enough to preclude reaching the islands by boat). On the other hand, the islands are usually well equipped with electricity (through undersea cables, or local generators), and 3G (soon 4G) coverage is good. Virtu Project, formed by a group of regional colleges of higher education[9], is part of the collaboration. The general aim is at the individual level, to help elderly in the archipelago area to continue living at home, support their social interaction, improve their quality of life and increase their safety.

The application for rehabilitation at home encompasses two aspects of health care and well-being: activity monitoring and activity feedback, integrated into everyday lives of senior citizens. The project does not only focus on the senior citizen of today, but also on the to-be senior citizen's expectations on applications designed to enhance their future everyday environments, thus enabling design solutions that will be sustainable both from a social and an economical/business point-of-view.

Our proposal consists of a remote physiotherapy monitoring system for rehabilitation. The software monitors exercise sessions for patients in rehabilitation after shoulder, hip or knee surgery. By using the Kinect sensor device and Kinect for Windows SDK (C#) we allow the patient to do the sessions at home giving feedback on the quality and frequency of the exercise to the physiotherapist expert remotely. The idea behind is that feedback should be twofold; it also should give the patient instructions on how to perform the exercise.

The application GUI uses Microsoft Kinect for Windows SDK (C#) as well as *Kinect toolbox*[10], although some tests were done also with *OpenNI* C++ (they use different skeleton representations). The application allows recording new patterns from different users realizing exercises for the system to learn recognizing them. Basic functionality is provided for the following features:

- Record and Replay: Records a session for training the system. Audio option activates and ends recording via voice (”Record”, ”Stop”).

- Stabilities: Indicates the degree of stability of the skeleton tracked.

- Capture and Delete Gesture: Adds (and deletes) a template gesture to a gesture-learning model.

- Capture T: adds a template posture to a posture learning model.

[9] http://www.virtuproject.fi/

[10] Kinect Toolbox: https://kinecttoolbox.codeplex.com/

- View Depth/View color: Shows depth/color image.
- Exercises to be trained and recognized in FRONT position with the camera:
  - Left and Right Hip Abduction
  - Left and Right Knee Extension
- Exercises in PROFILE position with the camera:
  - Left and Right Hip Extension
  - Sit and Stand.

Use case: "Five times sit to stand test"

An application example to demonstrate the architecture and ontologies developed is the *five times sit to stand test*[11]. Metaanalysis results demonstrated that "individuals with times for 5 repetitions of this test exceeding the following can be considered to have worse than average performance" [36].

- 60-69 y/o 11.4 sec
- 70-79 y/o 12.6 sec
- 80-89 y/o 14.8 sec.

To start a session the user should touch his head once until he hears the tin sound once, while to stop the session, he should touch his head again once until hearing the tin sound twice. The RESTful API of Smart-M3 works through a client that provides REST SPARQL requests and returns a JSON response. The architecture structure is in Figure 4.11, and some examples of possible SPARQL queries that can show triple level information associated to a sit to stand session are in Figure 4.12. As a result, the heterogeneous sensor integration from Philips Health Labs (PHL) store to Smart-M3 allows to obtain context-aware long-term evolution/changes in the patient [63].

[11] Five times sit to stand test `http://web.missouri.edu//~proste/tool/5x-STS.rtf`

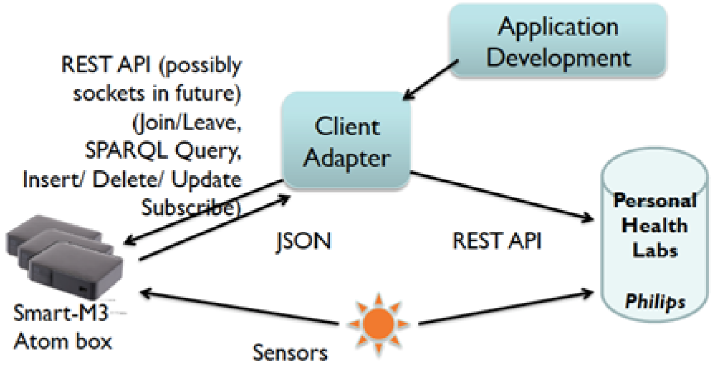
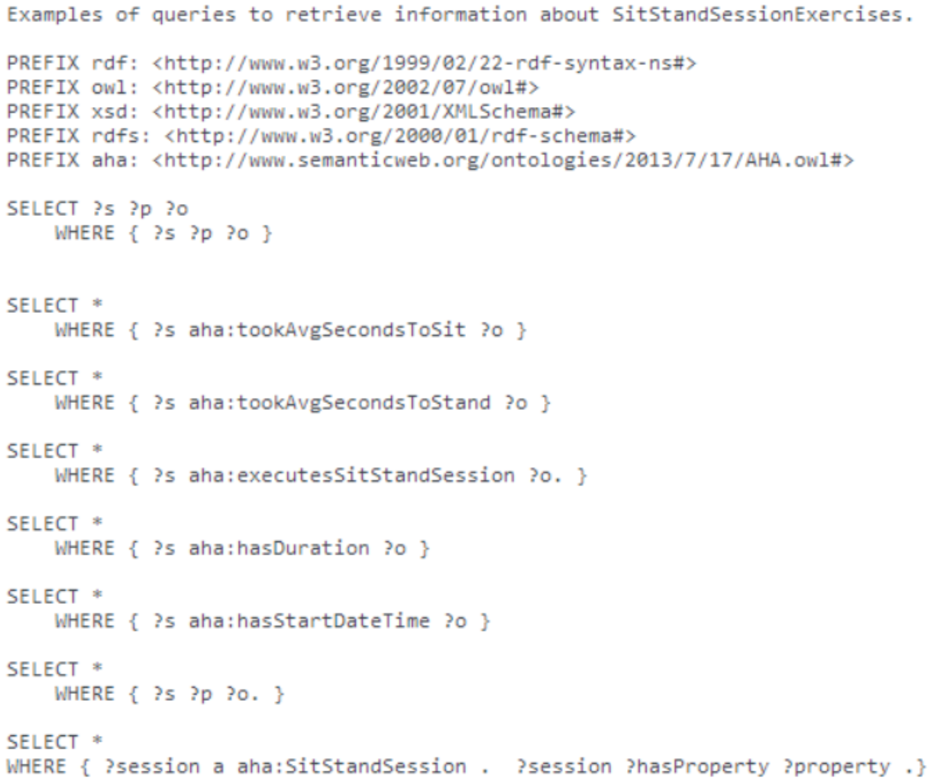


Figure 4.11: M3 low-power architecture for the remote rehabilitation system prototype [63] within *Active Healthy Ageing* platform project.



```
Examples of queries to retrieve information about SitStandSessionExercises.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX aha: <http://www.semanticweb.org/ontologies/2013/7/17/AHA.owl#>

SELECT ?s ?p ?o
    WHERE { ?s ?p ?o }


SELECT *
    WHERE { ?s aha:tookAvgSecondsToSit ?o }

SELECT *
    WHERE { ?s aha:tookAvgSecondsToStand ?o }

SELECT *
    WHERE { ?s aha:executesSitStandSession ?o. }

SELECT *
    WHERE { ?s aha:hasDuration ?o }

SELECT *
    WHERE { ?s aha:hasStartDateTime ?o }

SELECT *
    WHERE { ?s ?p ?o. }

SELECT *
WHERE { ?session a aha:SitStandSession .  ?session ?hasProperty ?property .}
```

Figure 4.12: Examples of SPARQL queries to obtain detailed information about the *sit to stand* session.

Chapter 5

Tackling uncertainty, vagueness and imprecision with a fuzzy ontology

> Uncertainty is an uncomfortable
> position, but certainty is an
> absurd one
>
> Voltaire

So far we have been tackling activity modelling with crisp (classical) ontological approaches. In this chapter we will approach the treatment of uncertainty, vagueness and imprecision in the way knowledge is represented. As human nature is not deterministic and humans perform complex behaviours, uncertainty, incompleteness and vagueness are unavoidable aspects to consider when modelling human activities. These are inherent components to the human behaviour and need to be accounted for. There exist distinctions among uncertainty versus vagueness, such as the treatment of gradual (or vague) propositions *in the presence of complete information*, in contrast to the handling of uncertainty for propositions. Likewise, there are differences in between possibility theory, probability theory and multiple-valued logics [75]. However, in this thesis, we consider a possibility theory fuzzy-logic based approach to compactly deal with all these aspects.

The previous case study 1 on a human activity recognition ontology is taken as starting base, and we add support to treat uncertain, vague and incomplete information. Next sections explain the required fundamental notions in order to achieve it.

5.1 Fuzzy ontologies

In many scenarios, and particularly in the human behaviour representation domain, we find elements whose nature is imprecise. A classic crisp ontology cannot represent this type of information, since they can only model relations between entities that may be either true or false. For instance, in the statement "User hasEvent Event planned in Location L at Time T", T does not have to be exact in practice. In another example, "User isPerforming CoffeeBreak", the activity *CoffeeBreak* could be recognized with some degree of truth depending on the sensor data acquired and how the user is performing the activity. Fortunately, fuzzy and possibilistic logic have proved to be suitable formalisms to handle imprecise/vague and uncertain knowledge, respectively [142, 203, 31]. Contrary to classical set theory, where elements either belong to a set or not, in the fuzzy set theory, elements can belong to a set with some degree. Formally, a fuzzy subset A of X is defined by a membership function μA(x), or simply A(x), which assigns any x∈X to a value in the real interval between 0 and 1. Fuzzy logic allows to perform approximate reasoning involving inference rules with premises, consequences, or both of them containing fuzzy propositions [33].

Up to 17 formal definitions can be found for fuzzy ontology [31]. However, one of the most accepted definitions is *an ontology that uses fuzzy logic to provide a natural representation of imprecise and vague knowledge, and eases reasoning over it.* Fuzzy Description Logic is the most developed formalism to work with fuzzy ontologies [31]. Formally, a Fuzzy Knowledge Base (FKB) or fuzzy ontology can be considered as a finite set of axioms that comprises a fuzzy ABox A and a fuzzy TBox T [34]. A fuzzy ABox consists of a finite set of fuzzy (concept or role) assertions, while a fuzzy TBox consists of a finite set of fuzzy General Concept Inclusions (Fuzzy GCIs), with a minimum fuzzy degree of subsumption. Fuzzy ontologies and fuzzy extensions of DL have great advantages versus crisp ones [61, 66]. They have shown to be useful in applications from information retrieval and image interpretation to Semantic Web and others [34]. In [100], a fuzzy keyword ontology serves to annotate and search events in reports by superimposing a fuzzy partonomy[1] on fuzzy classifications. They also have been used for reaching consensus in group decision making [167, 218], multi-criteria decision making [202], or extending information queries to allow the search to also cover related or incomplete results [160]. This results on more effective retrieval. Fuzzy ontologies have demonstrated to improve recognition accuracy by giving a more accurate degree of certainty, not compromising the recognition time in practice [61] with respect to crisp approaches. The combination of fuzzy logic and Semantic Web has shown to be useful in diverse

[1] A fuzzy partonomy is a decomposition that fuzzifies the classical whole-part relationship.

areas, e.g., in situation awareness for service [16], wine [41], or smartphone [150] recommendation among other applications.

In the area of activity recognition, fuzzy ontologies can define that the *CoffeeBreak* activity is recognized accounting for different weights on the sub-activities that compose it (e.g. 0.3 *TakeMug*, 0.3 *TakeCoffeePan*, 0.4 *TakeMilk*). Thus, when a sub-activity has been skipped due to an exception (e.g. milk run out) or a missing sensor reading, the activity can still be recognized to a lower degree. In contrast, the same activity formalized in a crisp ontology could not be recognized if any of the exclusive elements that compose it is missing.

Given a crisp ontology, elements that can be fuzzified include datatypes, object properties (through fuzzy modifiers), and data properties (through fuzzy modified data types). Furthermore, depending on the application domain, it can be frequent to have assertions of axioms about concrete individuals and classes, as well as (data and object) property assertions with a fuzzy degree. Among different approaches for fuzzifying an ontology, we can find Fuzzy OWL 2 mappings [33] or value discretization approaches [32]. To fuzzify a crisp ontology, it needs to be translated into a language supported by a fuzzy ontology reasoner. Fuzzy OWL 2 parsers convert Fuzzy OWL 2 ontologies into *DeLorean* [31] and *fuzzyDL* [33] reasoners' syntax. *DeLorean*[2] [31] is a fuzzy rough DL reasoner that supports fuzzy rough extensions of the fuzzy DLs $\mathcal{SROIQ}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ (equivalent to OWL and OWL 2) and it is based on a discretization of the fuzzy ontology using α-cuts [31]. *DeLorean* computes an equivalent non-fuzzy representation in OWL or OWL 2. However, we consider *fuzzyDL* to be the most convenient existing tool for ontological reasoning with uncertainty.

5.2 Handling automatic uncertainty reasoning with *fuzzyDL* reasoner

We consider *fuzzyDL* to be the most convenient existing tool [66] for ontological reasoning with uncertainty. *fuzzyDL*[3][33] reasoner's main features are the extension of the classical description logic $\mathcal{SHIF}(\mathbf{D})$ to the fuzzy case. It allows fuzzy concepts with left-shoulder, right-shoulder, triangular, and trapezoidal membership functions, general inclusion axioms and concept modifiers. Fuzzy modifiers apply to fuzzy sets to change their membership function. *FuzzyDL* supports crisp intervals that can serve to define fuzzy concrete predicates. In fuzzy rule based systems (e.g. Mamdani IF-THEN system), fuzzy IF-THEN rules are fired to a degree which is a function of

[2] *DeLorean* Reasoner: `http://webdiis.unizar.es/~fbobillo/delorean`

[3] *fuzzyDL* Reasoner: `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html`

the degree of match between their antecedent and the input. The deduction rule is generalized Modus Ponens. *FuzzyDL*'s reasoning algorithm [33] uses a combination of a tableau algorithm and a MILP (Mixed Integer Linear Programming) optimization problem.

By adopting fuzzy reasoners, we can accept axioms (of activities or behaviours) happening with a certain degree of truth, completeness or certainty. In *fuzzyDL*, the notion of satisfaction of a fuzzy axiom E by a fuzzy interpretation $\mathcal{I}$, denoted $\mathcal{I} \vDash E$, is defined in [33] as follows:

- $\mathcal{I} \vDash \langle \tau \geqslant \alpha \rangle$ iff $\tau^{\mathcal{I}} \geqslant \alpha$,
- $\mathcal{I} \vDash (\mathit{trans}\ R)$ iff $\forall_{x,y \in \Delta^{\mathcal{I}}}, R^{\mathcal{I}}(x,y) \geqslant sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,z) \otimes R^{\mathcal{I}}(z,y)$,
- $\mathcal{I} \vDash R_1 \sqsubseteq R_2$ iff $\forall x, y \in \Delta^{\mathcal{I}}.R_1^{\mathcal{I}}(x,y) \leqslant R_2^{\mathcal{I}}(x,y)$,
- $\mathcal{I} \vDash (\text{inv } R_1 R_2)$ iff $\forall x, y \in \Delta^{\mathcal{I}}.R_1^{\mathcal{I}}(x,y) = R_2^{\mathcal{I}}(y,x)$.

In *fuzzyDL*, concept C is satisfiable iff there is an interpretation $\mathcal{I}$ and an individual $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) > 0$ [33]. For a set of axioms $\mathcal{E}$, we say that $\mathcal{I}$ *satisfies* $\mathcal{E}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{E}$. $\mathcal{I}$ is a model of E (resp. $\mathcal{E}$) iff $\mathcal{I} \vDash E$ (resp. $\mathcal{I} \vDash \mathcal{E}$). $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, denoted $\mathcal{I} \vDash \mathcal{K}$, iff $\mathcal{I}$ is a model of each component $\mathcal{A}, \mathcal{T}$ and $\mathcal{R}$, respectively. $\mathcal{A}$ represents a fuzzy *ABox*, $\mathcal{T}$ a fuzzy *TBox*, and $\mathcal{R}$ a fuzzy *RBox*. $\mathcal{A}$ contains a set of fuzzy concepts and fuzzy role assertion axioms. $\mathcal{T}$ holds a set of fuzzy General Concept Inclusion axioms (GCIs) or instantiations of a given concept to a certain degree α. Finally, R is a finite set of role axioms that states the fact of a role being functional, transitive, or when a role subsumes, or is inverse, to another role.

An axiom E is a *logical consequence* of a knowledge base K, denoted $\mathcal{K} \vDash E$ iff every model of $\mathcal{K}$ satisfies E. Given $\mathcal{K}$ and a fuzzy axiom τ of the form $\langle x : C, \alpha \rangle$, $\langle (x,y) : R, \alpha \rangle$ or $\langle C \sqsubseteq D, \alpha \rangle$, it is of interest to compute τ's best lower degree value bound.

The *greatest lower bound* of τ w.r.t. $\mathcal{K}$ (denoted $glb(\mathcal{K}, \tau)$) is $glb(\mathcal{K}, \tau)$ = sup $\{n | \mathcal{K} \vDash \langle \tau \geqslant n \rangle\}$, where sup $\emptyset = 0$. Determining the *glb* of a concept C, i.e., the Best Degree Bound (BDB) problem, consists of determining the best satisfiability bound of a concept C:
$glb(\mathcal{K}, \text{C}) = sup_{\mathcal{I}} sup_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) | \mathcal{I} \vDash \mathcal{K}\}$

Reasoning tasks allowed by *fuzzyDL* are typical BDB, concept satisfiability and subsumption problems, optimization of variables and defuzzifications. For more details on *fuzzyDL* syntax and semantics, we refer the reader to [33, 31].

In the next section, we detail the components of our proposal on a crisp ontology for human behaviour modelling and how its fuzzification is performed by using *fuzzyDL*.

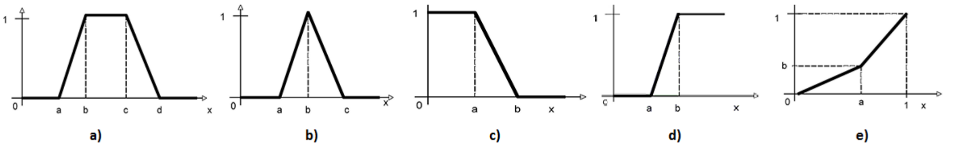


Figure 5.1: Membership functions in [31] for fuzzy data type definition (i.e., fuzzy concrete domains) and fuzzy modifier functions: a) Trapezoidal function; b) Triangular function; c) Left-shoulder function; d) Right-shoulder function; and e) Linear function.

5.3 Fuzzification of the human activity recognition crisp ontology

The formal specification of human behaviour is difficult to handle when regular crisp reasoning mechanisms are used for this purpose, since natural human patterns are imprecise, imperfect and fully gifted with semantics. Fuzzy ontologies and fuzzy extensions of Description Logics [24] arise as more appropriate formalisms to deal with the vagueness inherent to real-worlds domains [31].

To represent our fuzzy entities in the created human activity ontology, we use *Fuzzy OWL2 2.1.1* plug-in[4] in Protégé 4.1, jre 1.6, that provides support in creating *Fuzzy OWL 2* ontologies. The plug-in does not translate fuzzy representations into OWL 2, but rather eases their representation by allowing specification of the type of fuzzy logic used, definition of fuzzy data types, fuzzy modified concepts, weighted concepts, weighted sum concepts, fuzzy nominals, fuzzy modifiers, fuzzy modified roles and data types, and fuzzy axioms [34]. In *Fuzzy OWL 2*, three main alphabets of symbols are assumed: concepts (fuzzy sets of individuals), roles, and individuals [34]. These are represented in an ontology as classes, relations, and individuals, respectively. The degree of truth of a fuzzy assertion is equal to the proportion of observers who think that the crisp assertion is true [31].

Considering the designed human activity ontology, we can identify data types, concepts, properties, and relations that are susceptible of being fuzzy. A fuzzy data type **D** is a pair $\langle \Delta_D, \Phi_D \rangle$ where Δ_D is a concrete interpretation domain, and Φ_D is a set of fuzzy concrete predicates **d** with an arity n and an interpretation $\mathbf{d}^{\mathcal{I}} : \Delta_D^n \rightarrow [0,1]$, which is an n-ary fuzzy relation over Δ_D[34]. For fuzzy data types, the functions allowed in Fuzzy OWL 2, defined over an interval $[k_1, k_2] \subseteq \mathbb{Q}$, are $\mathbf{d} \rightarrow \{\ left(k_1, k_2, a, b)$ (Fig. 5.1c), $right(k_1, k_2, a, b)$ (Fig. 5.1d), $triangular(k_1, k_2, a, b, c)$ (Fig. 5.1b), $trapezoidal(k_1, k_2, a, b, c, d)$ (Fig. 5.1a), $linear(k_1, k_2, c)$ (Fig. 5.1e), $mod(\boldsymbol{d})\ \}$. More specifically, the fuzzification of each element in the ontology

[4] http://nemis.isti.cnr.it/~straccia/software/FuzzyOWL/#plug-in

is done as follows:

- **Fuzzy Data Types and Fuzzy Concrete Roles (data properties)**. Data properties in the original ontology can be transformed into fuzzy data types. Their range is expressed through data range expressions such as (*double*[>= −100.0] and *double*[<= 100.0]) (e.g. for *hasTemperature* data property range). We use as range the referential set over which the fuzzy membership functions associated are defined. On the other hand, fuzzy concrete role is defined in Fuzzy OWL 2 by setting its range data type to a previously defined fuzzy data type. Examples of membership functions used are:

  - *LowTemperature*: fuzzy data type with left shoulder membership function (a= -5, b = 5).
  - *MediumTemperature*: fuzzy data type with trapezoidal membership function (a= 5, b = 10, c = 20, d = 25).
  - *HighTemperature*: fuzzy data type with right shoulder membership function (a= 25, b = 30).

  An annotation example in OWL 2 for the data type *highTemperature* is as follows:

```
<fuzzyOwl2 fuzzyType="datatype">
<Datatype type="rightshoulder" a="25.0" b="
    30.0" />
</fuzzyOwl2>
```

Listing 5.1: Annotation for a new fuzzy data type *highTemperature* using rightshoulder modifier

  When creating a fuzzy role, an annotation property describing the type of the constructor and the value of its parameters are specified. Recursion is not allowed in the definition, and following the mapping in [34], only fuzzy modified roles are supported. The domain of the annotation will be any OWL 2 (object or data) property with the restriction that the modifier must be defined as a fuzzy modifier and that the base fuzzy role has a different name than the annotated role. Examples of fuzzy data types defined in our ontology are:

  - *ShortDuration, MediumDuration, LongDuration* are fuzzy data types (FDT) used to represent duration (in seconds). The concrete role (CR) Activity.*hasDuration* indicates the duration of an *Activity.*
  - The FDT *LowVolume, MediumVolume, HighVolume* are used to represent audio volume level in dB. The CR Device.*hasVolume*

indicates the volume of a *Device* with audio capability (computer, radio, TV...).

- The FDT *ClosedAperture, HalfAperture, OpenAperture* describe aperture angle in degrees of the CR {Door, Window, Curtain, etc.}.*hasAperture.*
- *SmallCapacity, MediumCapacity, LargeCapacity* are FDT representing the amount of people, and they as used in CR of Location.*hasCapacity* to describe the capacity of a *Location.*
- The FDT *LowHumidity, MediumHumidity, HighHumidity* are used to model humidity (in g/m^3) in the CR Location.*hasHumidity* of a *Location.*
- *LowTemperature, MediumTemperature, HighTemperature* are FDT to measure temperature in centigrade degrees, and they are applied over CR {Location, Room, Environment, GenericUser, Object}.*hasTemperature*) of a *Location, Environment, User, or Object.*
- *LowLighting, MediumLighting, HighLighting* are FDT measuring lighting in lux, applied to CR of {Location, Environment}.*hasLighting* of a *Location* or *Environment.*
- The FDT *LowNoiseLevel, MediumNoiseLevel, HighNoiseLevel* are designed to measure noise level in dB in the CR {Location, Environment}.*hasNoiseLevel* of a *Location* or *Environment.*
- *LowPressure, MediumPressure, HighPressure* are FDT to measure pressure in atmospheres, and they are used in the CR of {Location, Environment}.*hasPressure* of a *Location* or *Environment.*
- *FewPeople, MediumPeople, ManyPeople* are integer-valued FDT to describe a number of people, in the CR of Event.*hasNAttendants*) such as *Conference, Seminar, Symposium, Workshop* or other *Event.*

- **Fuzzy Abstract Roles** (Fuzzy Object Properties). Object properties in the original ontology can be transformed into fuzzy abstract roles by means of the assignment of a fuzzy membership value. Examples in our ontology are:

  - Thing-*isInLocation*-Location: Represents the location of anything (a *Thing*) in a given *Location.* A fuzzy degree can represent proximity.
  - User-*attendsEvent*-Event: Identifies a given *User* who attends an *Event.* Also, *Event* can belong to a *Calendar.*

  - Travel-*toLocation*-Location: Characterizes the activity *Travel* by specifying the destination *Location* it refers to.
  - Activity-*happensInLocation*-Location: Associates an *Activity* with a *Location* to indicate where it occurs.
  - Action-*actionAppliesTo*-Thing: Indicates the object over which an *Action* or *Activity* directly falls or acts over. e.g., *WalkBy* activity applies to *Corridor* if a user walks by a corridor.
  - GenericUser-*hasPersonalStatus*-PersonalStatus: Indicates a personal status of any *User* in certain moment. *PersonalStatus* class is specialized into: *Available, Away, Busy, OnHoliday, OnLeave.*
  - Activity-*involvesAction*-Action (and *involvesActivity, involvesOptAction, involvesOptActivity,* respectively, where *Opt* stands for optional, relates an *Activity* with the *Actions* that it involves or requires. e.g., the *DoPresentation* activity involves actions *SetLaptopOn*, *SetProjectorOn*, *StandUp*, and *Talk*.
  - GenericUser-*performsAction*-Action and GenericUser-*performsActivity*-Activity specify which *User* performs an *Action* or an *Activity*, respectively. A fuzzy degree here represents the level of uncertainty about who performs the action.
  - Thing-*isNearTo*-Thing describes closeness among two entities (*Object, User, Location,...*).

- **Fuzzy Modifiers and Fuzzy Modified Data Types**: The degree of membership of fuzzy data types may be specialized by means of fuzzy modifiers. A fuzzy modifier is a function $f_{mod} : [0,1] \rightarrow [0,1]$ which applies to a fuzzy set to change its membership function, which can be *linear(c)* (Fig. 5.1e) or *triangular(a, b, c)* (Fig. 5.1b). We define the fuzzy modifiers *very* as *linear(0.85)*, and *barely* as *linear(0.15)*, to improve the expressiveness of the ontology. For instance, given the object property *isNearTo(Thing, Thing)*, the fuzzy modifiers can be used to define new fuzzy properties such as *very(isNearTo)* or *barely(isNearTo)* (i.e., *isVeryNearTo*, or *isBarelyNearTo*), and relate objects that are not close to each other, but *very* (respectively *barely*) *close*[5].

- **Fuzzy Axioms or Assertions**: In practice, expressing degrees of truth is reflected on real time when asserting axioms. Examples of fuzzy axioms can be assertions such as the following example: the *User* individual *John IsInLocation Office* with degree 0.7, if the system has detected usual activity of John at his office, but there are some changes in his routine).

[5] Strictly, *fuzzyDL* does not support yet modifiers applied to roles, but to concepts only. However, Fuzzy OWL 2 allows to apply modifiers to roles as well

When asserting axioms and defining data properties, linguistic labels make the editing of the rule closer to natural language and also easier for any kind of user, not requiring any technical knowledge. A degree within the interval (0, 1] can, in addition, be provided for more precise assertions or axioms. For instance, to find whether there is any user which is *close enough* to the meeting room, we could assign a certainty value >0.8 to the query. Fuzzy queries will be provided in the experimental section as a proof of concept and for ontology evaluation.

5.3.1 Fuzzy reasoning

Our fuzzy ontology is mainly characterized by the presence of fuzzy data types, which are especially handled by *fuzzyDL*. For this reason, we use this reasoner in our ontology. We do not specify degrees of truth initially on the ontology, these are asserted in execution time. Thus, α-cuts cannot be pre-calculated as in the *DeLorean* reasoner. *FuzzyDL*, however, has some limitations that are analysed in this section, as its implications in our proposal.

First, *fuzzyDL* does not allow representation of asymmetry and irreflexivity role axioms with the available constructors. In practice, this entails no critical implications to our work, since our roles are defined with domain and range for each property, and these are usually disjoint classes. Therefore, while it is possible to express reflexive and symmetric roles in *fuzzyDL*, we need to make sure that there is not any asymmetry or irreflexivity, since they cannot be enforced by the reasoner. Secondly, cardinality restrictions are not implemented in *fuzzyDL*. This problem may cause two situations that we solve as follows: 1) e.g., in the *Event* class definition, *belongsToCalendar min 1 Thing* is equivalent to *belongsToCalendar some Thing*; and 2) In the *User* class definition, *hasNUsers exactly 1 int* can be substituted by *hasNUsers some int*. This fact, in addition to having *hasNUsers* as functional property, makes the number of relations *hasNUsers* exactly one.

In general, activities are composed of a sequence of actions. However, this does not happen in practice all the time, e.g., some cycles can appear to complete "half done" actions, or forgotten actions can be executed only when the person has realized of the oversight, at the end, after the rest of actions. Some behaviour *procedures* are not followed in the logical order, but in the order in which the actions are actually necessary, during its execution, without any specific sequence ordering (e.g., taking instruments while cooking). Since each behaviour has not a unique way to be performed, behaviours can be specified by a set of actions and/or activities that compose it. This is done through specifying the time of occurrence of actions, activities, and behaviours, and a membership function that represents the belonging of the action/activity to a given behaviour. Regarding the time of occurrence, actions have a crisp datetime as a timestamp. However, activities and actions have a *startDatetime, endDatetime*, and *duration.* Duration is a fuzzy datatype given by a triangular fuzzy function. Therefore, when an activity or behaviour is recognized, it will be with a degree of truth, and it will be logged as an activity or behaviour detected with a membership value in (0, 1]. e.g.: The behaviour *GoingForAWalk* could have been recognized with an associated degree: *(Natalia:GoingForAWalk, 0.8).*

5.3.2 Case study 2. A fuzzy ontology for human activity recognition in the work/office domain and public buildings

Offices and public buildings are a good scenario to test our approach, since the inclusion of the designed ontology into an automatic building control system could help to improve both energy efficiency and occupants' wellness. This section describes a proof of concept about the potential of the fuzzy ontology in this domain. Later, in Chapter 7, the approach will be evaluated, and an evaluation and comparison with crisp ontologies to show the benefits of the approach will be shown. The main elements that are used to model knowledge in the office university environment are the following:

- User status. At work a user can have different statuses such as away, at work, on leave, on holiday, busy, or available. Users can also have work positions (secretary, researcher, technical staff, students, lecturers, etc.). Generic users in this domain can be company representatives, internal representatives, research partners, visitor researchers, visitor students, technical staff, etc.

- Physical environment. The architectural environment is also represented in order to monitor activities through presence sensor or cameras. Part of the environment are locations and their corresponding GPS coordinates. For this reason, we allow the representation of semantic maps including different components of indoor locations such as different types of buildings (public and privates) and different specializations of rooms (lecture rooms, laboratories corridors, kitchen, etc.), and outdoor locations such as parks, terraces, and means of transport.

- Objects. Any element subjected to possible interaction can be considered in this category. From furnishings such as windows and doors to desktop tools and devices. These devices connect to a specific network (that provides the context), and can represent a role such as addressee or sender devices in a given communication.

- Actions. Actions considered are not only focused on the work environment (setting projectors on/off, turn devices on/off, check-in at location, etc.) but also include those that can be considered in other environments (open fridge, being observed at location, etc.).
- Activities. Activities in the work environment include general and specific tasks such as going to work, lectures, meetings, presentations, writing, etc.
- Behaviours. Behaviours can be defined for specific routines or patterns based on actions, activities, and other environmental conditions. For instance, it is possible to express relations of interest including *working for a certain project, being a partner, colleague (lab, department, university), advisor*, etc. Modelling from general behaviours related with events and calendars, to in-office activities and object interaction is a possibility. When there is a lack of context information, but there exists evidence of activities happening, we may reason with *UnknownUser(s), UnknownLocation, UnknownActivity* or *Unknown-Behaviour.* In these cases, other context information within the same time frame may be crucial to help disclosing the human behaviour.

The current version of the developed ontology, validated with the OWL reasoners HermiT 1.3.6, Pellet 2.3.0 (2.2.0 Protégé plug-in) and *fuzzyDL 1.1*, consists of 228 classes, 133 object properties, 62 data properties and 33 test individuals, within $\mathcal{SROIQ}(\mathbf{D})$ DL expressiveness. The overall ontology can be seen in Figure 5.2.

Pitfalls found in the ontology modelling process were corrected using the OOPS! Pitfall Scanner [164]. To put an example of behaviour definition, we can identify the actions/activities that compose it and model it with timing ordering relations. Next, some examples show applications of use of our ontology:

Example 1**: Defining a Behaviour.** The behaviour "having a coffee break" can be composed of a set of actions: exiting the office, going through the corridor to the kitchen, taking coffee if somebody already made it or making coffee if there is not any left, and then coming back to your office. In ontology terms, this could be expressed with a nested hierarchy of actions and two activities, *MakeCoffee* and *TakeCoffee* (underlined):

- Behaviour *To have a coffee break*: OpenDoor WalkBy(Corridor) BeObservedInLocation(Kitchen) (MakeCoffee + TakeCoffee) WalkBy(Corridor) OpenDoor.

-Activity *MakeCoffee*: OpenCupboard TurnCoffeeMachineOn MoveObject(CoffeeJar) OpenFridge

-Activity *TakeCoffee*: MoveObject(CoffeeJar) OpenFridge

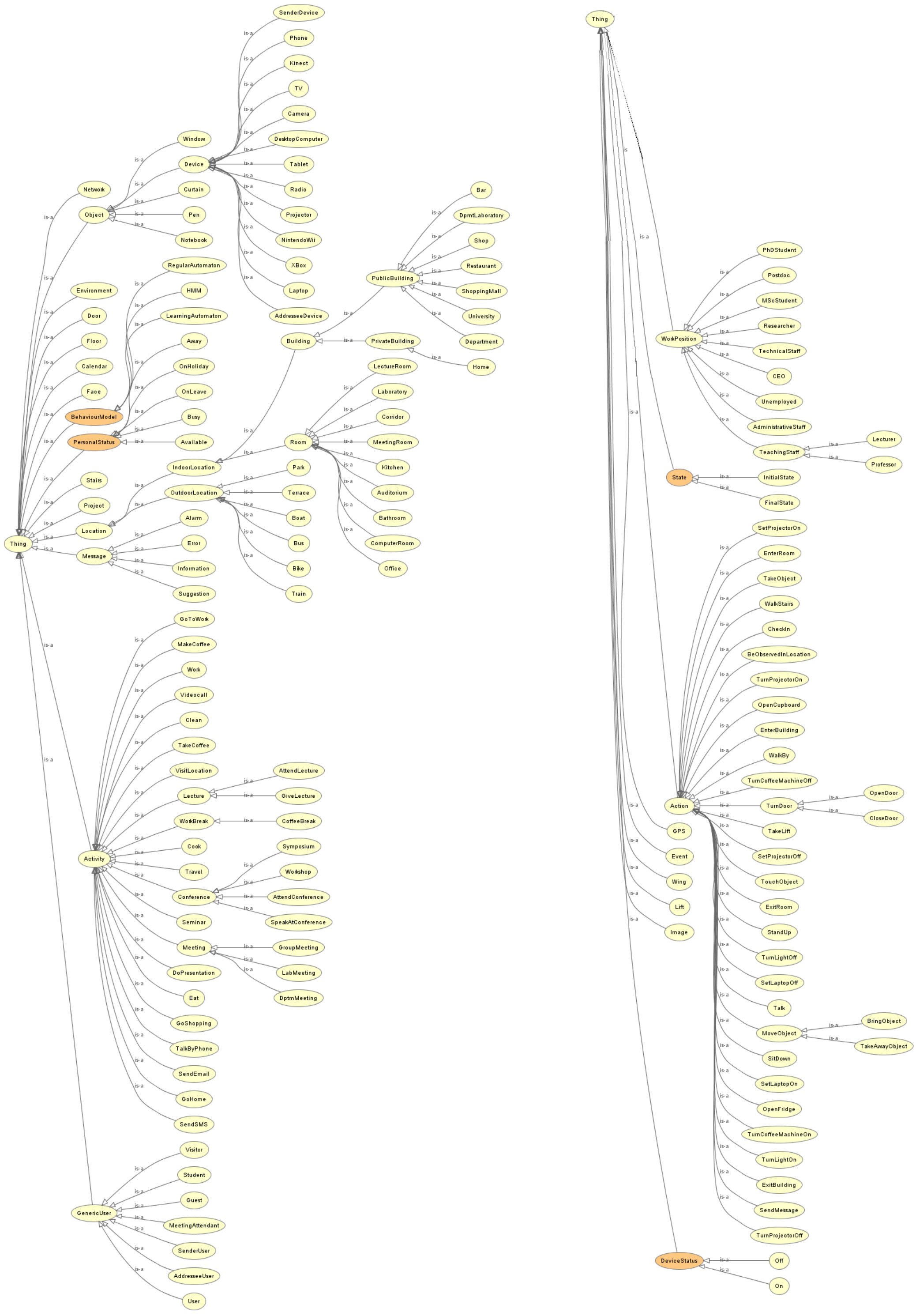


Figure 5.2: Full human activity recognition ontology developed for the work/office domain [61].

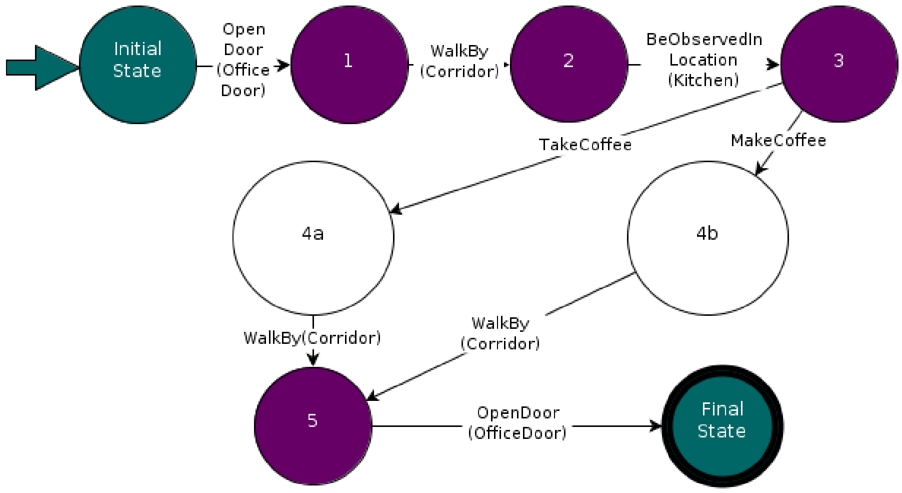


Figure 5.3: Underlying regular automaton to model the behaviour *To have a coffee break.*

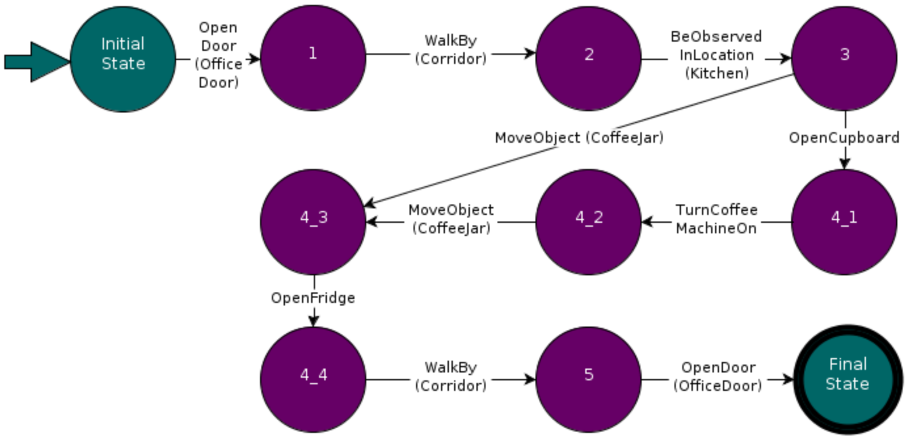


Figure 5.4: Underlying regular automaton (at lower-Action level) to model the behaviour *To have a coffee break.*

Figure 5.3 shows the underlying state machine to recognize the same behaviour. Activities are represented with white large nodes, while actions are represented with small purple nodes. Activities are, at the same time, abstractions of other state machines able to recognize fine-grained activities. Figure 5.4 shows, in more detail, the same behaviour at action level. This hierarchical and recursive design is useful to decompose the recognition process into different levels of granularity or abstraction. As an example, the activity *MakeCoffee* could be expressed with the following axioms:

$$\forall\ U \in User : performsAction(U, OpenCupboard) \wedge performsAction\ (U, TurnCoffeeMachineOn) \wedge performsAction(U, MoveCoffeeJar)\ \wedge performsAction(U, OpenFridge) \rightarrow performsActivity(U, MakeCoffee).$$

***Example 2*: Defining a behaviour with OWL 2 axioms.** Let us suppose we want to define a meeting with the vice-chancellor or Rector as a special type of meeting. The concept *HavingRectorMeeting* can be defined as a meeting in any auditorium or meeting room, which is hosting at least 10 people and someone of those room occupants has the *Rector* work position. *HavingRectorMeeting* will be a specialization class of *Meeting*; more concretely, with the following restrictions:
HavingRectorMeeting = Meeting and (happensInLocation some ((Auditorium or MeetingRoom) and (isLocationFor some (hasWorkPosition some Rector)) and (isHostingNPeople some int [>="10"^^int]) and (hasProjector some (hasDeviceStatus value On))))

***Example 3*: Defining Crisp Rules.** Concrete application domain rules can be modelled in pure OWL 2, Fuzzy OWL 2 (e.g. with Mamdani rules) or rule languages such as SWRL [swr] or SPIN [spi]. For instance, if there is a scheduled conference for more than 25 people, we can automatically activate the lights 10 minutes before the event starts. Let L be a variable for any *Location*, X be an integer variable, and T be a timestamp. Then the rule can be modelled as follows:

$$\forall E \in Event, L \in Location, \forall C \in Conference, X \in int, T \in datetime : isProgrammedWithinEvent(C, E) \wedge isScheduledAtLocation(E, L) \wedge hasNAttendants(E, X) \wedge X > 25 \wedge hasProgrammedStartDateTime(E, T) \rightarrow atDatetime(TurnLightsOn, 'T - 00:10:00') \wedge lightsAtLocation\ (TurnLightsOn, L)$$

TurnLightsOn is an example of external applications with service grounding. They are modelled as subclasses of the *Service* class and their associated data or object properties represent the application parameters.

***Example 4*: Executing Fuzzy Queries.** There are several ways of running a query. One of the aims of running a query is finding if a rule is triggering. For instance, to determine the minimal degree to which individual *NataliasNokiaN8* is an instance of concept *Phone*, we would run: *(min-instance? NataliasNokiaN8 Phone)*. However, if we want to know all *Phone* concept instances, the following query applies the previous query to every individual in the KB: *(all-instances? Phone)*.

On the other hand, if we want to determine the maximal degree to which individual pair *(Natalia, JohanLiliusOffice)* is an instance of role *isInLocation*, this can be obtained as: *(max-related? Natalia JohanLiliusOffice isInLocation)*. In the opposite side, another possible query is finding the minimal degree to which a concept A, e.g., *VeryFullCapacity*, subsumes a concept B, e.g., *FullCapacity*. This query would be expressed as: *(min-subs? VeryFullCapacity FullCapacity)*. Optionally, an individual, as well as Lukasiewicz, Gödel, or Kleene-Dienes implications, can be used for this type of query [33].

***Example 5*: Defining Fuzzy Rules.** Fuzzy rules in *fuzzyDL* can be expressed with the Mamdani structure or as implication rules. These can be mapped to a set of statements in a fuzzy KB as a fuzzy control system [33]. The definition of logical rules as Mamdani rules is performed as follows: *(define-concept MamdaniRuleBase (g-or Rule1 (...) RuleN))*. For example, the rule "If the User Natalia is away for the weekend and the weather situation in Turku becomes very stormy, all electricity appliances should be turned off" can be expressed as: *(define-concept Rule1 = (g-and (Natalia (some hasStatus AwayForWeekend)) (WeatherSituationTurku (some isCurrently VeryStormy)) (TurnOffAllElectricitySwitches (some withParams NataliasAppartment))))* .

The input to the controller/facts can be done using the following syntax: *(instance input (and WeatherSituationTurku (some isCurrently NearlyCloudy))) (instance input (and Natalia (some hasStatus AtWork)))* (...).

In addition to the previous example, we may be interested in finding the real value of a fuzzy concept when using different fuzzy linguistic labels. This process is known as defuzzification and it can be done with the following command for the previous rules: *(defuzzify-lom? MamdaniRuleBase input TurnOffAllElectricitySwitches)*.

Alternatively and equivalently, the definition of logical rules can be done as implication rules instead. In this case, we show how to encode a rule that detects if Natalia's phone is in a location near Johan's office. If this happens, it is recognized that they are having a meeting and starts recording in her phone the agenda and transcribing it from her phone to her calendar to have it into account for the next meeting:

(define-concept antecedents (and (Natalia (some hasPhone P) (and (Natalia (some hasCalendar C)) (and (P (some isInLocation L))) (and L (some isVeryNearTo JohansOffice))))))

(define-concept consequents (and (StartAudioRecording (some withParams P)) (TranscribeMeetingAgenda (some withParams (and P C))))))

(define-concept Rule2 (l-implies (g-and antecedents consequents)))

The query for the consequent' satisfiability degree could be carried out with the query: *(min-instance? input consequents)*.

This chapter concludes the ontology design contribution part of the thesis. The following chapter will present the second largest contribution, i.e., a hybrid architecture for dealing with real-time tracking and recognition of human activity. Both the fuzzy ontology proposed and the hybrid semantic architecture contributions are evaluated in Chapter 7.

Chapter 6

Architecture proposal: A hybrid data-driven and knowledge-based ontological system for hierarchical and real-time activity recognition

> You're only given a little spark
> of madness. You mustn't lose it
>
> Robin Williams

Human activity recognition in everyday environments has been proved to be a critical task in Ambient Intelligence applications to achieve proper Ambient Assisted Living. Key challenges still remain to be solved to achieve robust methods. After studying in Sections 3.1 data-driven approaches to AR, and in Section 3.2.1, ontological approaches to AR, we take the best sides from both approaches in order to have a more robust and context-aware activity tracking and recognition system that is able to give meaning to the activities recognized, according to context. Our work is motivated by a lack of proposals tackling motion detection at the same time as figuring out the actual meaning of these activities, taking imprecise context into account. In addition, by having context into account and proper dealing of uncertainty, we will avoid the need for retraining the system when new input data or models change, according to context.

We will apply our proposed fuzzy ontology [61] from previous chapter, on top of a data-driven sub-activity recognition system, to give support for semantic interpretation, logic reasoning and management of imprecision and uncertainty in activity recognition scenarios. Our rationale is that a sensor can provide readings with a certain degree of reliability, or sample data only at specific times or under certain conditions; users may perform subtle changes in the way they perform their activities; the execution of an activity may be detected with uncertainty or a satisfiability degree; and the performance of an atomic action (for example, to take a dish) depends of the goal or intention of the user. In this last case, semantic interpretation of human actions are key to achieve a suitable activity recognition. As a consequence, all this information should be taken into account into the reasoning and recognition process.

The general proposed framework consists of two main modules (see Figure 6.1): the low level sub-activity recognizer and the high-level activity recognizer. Each module has been developed using different techniques: a data-driven approach for the first module and a knowledge-based context-aware one for the later. The first module detects sub-activities (actions or atomic activities) that take input data directly from sensors. In our implementation we have used Dynamic Time Warping to learn and recognize these sub-activities, which is classified as a data-driven method.

On the other hand, the second component lays on top of the previous one, in a superior level of abstraction. It gets input data from the first data-driven component (i.e., sub-activities) and executes ontological inference to provide with semantics both the activities and their influence in the environment. This component is thus knowledge-based, and we have used a fuzzy ontology to model high-level activities. In the following subsections we provide the details to show how the whole system is able to detect and recognize complex human activities as ADLs using an RGB-D sensor.

6.1 Phase 1: Data-driven modelling and recognition of low-level activities

In our activity recognition approach, we receive data from an RGB-D camera. By means of this sensor, we are able to extract data about the body postures of the users appearing in front of the sensor, as well as the 3D location of the objects in the scene. We focus on the time complexity of how to process the data received by this sensor efficiently, and how to make the system capable of understanding the user's whole activity in real time, to get closer to real-time applications in daily life. The device selected for this task is the Microsoft's Kinect camera, which offers depth and RGB images of the scene.

Our goal is to create a data-driven framework that learns, detects, and recognizes different events performed by users at their homes. Each one of these events or sub-activities will correspond to a sequence of images

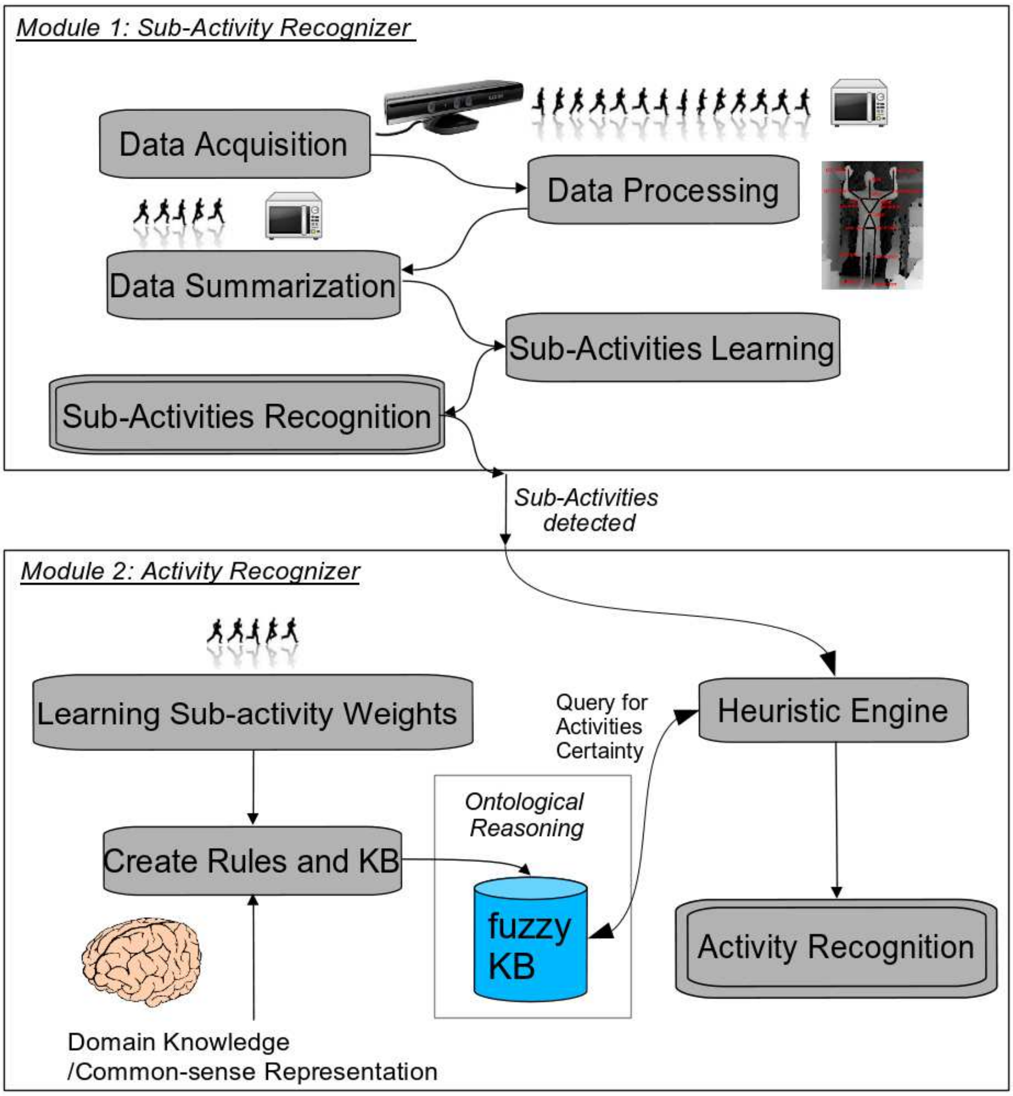

Module 1: Sub-Activity Recognizer
Data Acquisition
Data Processing
Data Summarization
Sub-Activities Learning
Sub-Activities Recognition
Sub-Activities detected
Module 2: Activity Recognizer
Learning Sub-activity Weights
Query for Activities Certainty
Heuristic Engine
Ontological Reasoning
Create Rules and KB
fuzzy KB
Activity Recognition
Domain Knowledge /Common-sense Representation

Figure 6.1: General diagram of the proposed hybrid framework.

obtained by the camera, in this case, a sequence of the user's body postures and the objects that take part in the sub-activity. The learnt sub-activities will be used in the upper layer to detect and recognize more complex human activities such as ADLs (involving one or more of these sub-activities). Once the data is acquired from the sensor, the sub-activity recognition framework comprises four additional steps:

- **Step 1: Initial processing of the 3D data provided by the camera**

  1.a: User's posture detection. We obtain the user's body skeleton represented as a set of fifteen 3D points (head, neck, torso and left and right shoulders, elbows, hips, hands, knees, and feet) through the camera and the middleware used. Afterwards, we process this data to represent the user's skeleton in a manner that is independent of the angle and distance to the camera. This representation will be the set of angles conformed by the user's body joints (angles between bones), plus the measured height value of the user's chest from the floor. In the end, the body posture on each frame received by the camera is represented by a set of eleven float values (ten angles plus the *height*).

  1.b: Object detection and tracking. The detection and tracking of objects in the scene must be carried out to infer and detect sub-activities involving their use. There already exist solutions to this problem as [108], and we have used the same implementation as in [128]. The process consist of: a) learning the features of a set of objects that can be placed in the scene. In this work, we acquired the objects included in the CAD-120 dataset and learned the RGB-D object dataset by Lai et al. [131]; b) the 3D bounding boxes of the objects detected, with a score above a threshold, are obtained; c) the object tracking can be done using the particle filter tracker implementation provided by the PCL library (Pointclouds library[1]). The result is a list of objects detected at every frame with a given 3D position of the centroid.

- **Step 2: Compressing the data time series.**

  In our problem, the time series data is the sequence of postures performed by the user and the objects positions. We summarize the sequence of postures in order to work with smaller amount of data. The method selected for this task has been the *Piecewise Aggregate Approximation* (*PAA*) [119].

- **Step 3: Training the model for sub-activities learning.**

[1] `http://pointclouds.org/documentation/`

The objective of this phase is to learn different sub-activities involving the use of objects by the user, for its later recognition. To this end, we use instance-based learning using k-Nearest Neighbours algorithm, where we have an instance database and each instance is labelled with an activity. We select a subset of instances of the recorded sub-activities in a training dataset as template activities, so that new instances acquired from the sensor are compared to the templates. The distance measure to compare these time series instances is the algorithm Dynamic Time Warping (DTW) [57], which measures similarity between two sub-activities and is invariant to their length, lags and speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other or if there were accelerations and decelerations during the course of an observation. The DTW time complexity is $O(m{\cdot}n)$, where m and n are the length of the time series instances. We selected this technique for training and recognition of activities because it provided the best results in both accuracy and time complexity in a preliminary experimentation. As an example, the work published in [134] describes a comparison of DTW with HMM, where we conclude that the use of the proposed framework with DTW results in a faster execution time and recognition accuracy.

- **Step 4: Sub-activities recognition.**

  Finally, once we have the different models trained, the system will be able to detect and recognize a new sub-activity sample performed by the user. The usual operation is that, after the depth sensor produces a new RGBD image sample, this is used as input for steps 1-2 explained above and then, compared within the different trained models of step 3. Each one will return an output score, reflecting the similarity of the input sequence that conforms the model. The instance in the database with the best membership score will be classified as the sub-activity being performed.

Algorithm settings and feature selection for sub-activity recognition

This section describes the features used for recognition of sub-activities after the data are acquired from the RGB-D sensor in depth. In summary, we use two types of features: Those for user detection, and features for object affordance recognition.

1. **Skeleton features.** Once the framework has been implemented, the next step was to configure it before the experiments take part. Since

we are using the CAD-120 dataset, some preliminary tests were done to see if the features selected initially were valid. We realized that the representation of the whole skeleton from the user was introducing some noise, as in most of the activities only the upper body was used, and sometimes the lower body was occluded. This problem is first motivated by the course of dimensionality, since motion is a multivariate time series. Therefore, finally only the set of angles representing the upper body were selected as features. These are four angles, representing the user's arms independently from the camera position (see Table 6.1). This is actually an advantage, as we are reducing the amount of data to be computed, easing our aim of reaching real-time recognition. Future work will focus on making this feature selection automatically.

2. **Objects features.** As shown in Table 6.1, 16 object features will be used as part of the global features vector used for the DTW algorithm. The first restriction we applied is that objects are only considered when the Euclidean distance between the object and any of the user's hand is less than 40 centimetres. While this happens, we keep track of the object by including its distance to the hand as part as the features vector. This information is saved in different manners depending on the type of object and its identifier as follows:

   (a) if the object type cannot be found repeated within the same activity (e.g. there are not two *microwaves*), its distance will only be saved once in the appropriate cell (from 1 to 10, as there are 10 different object types) of the features vector related to the objects type (see Table 6.1).

   (b) if the object type can be found repeated within the same activity (e.g. *stacking* several bowls) we use an object identifier, and its distance will be saved once in the appropriate cell (from 1 to 5, as the maximum number of repeated objects is 5) of the features vector related to the objects identifier (see Table 6.1).

   Finally, we also want to represent the relationships between objects appearing in the global scene in another feature (see Table 6.1). To this end, we compute the sum of the Euclidean distances between all objects that take part on the activity, representing the objects movements.
   With these assumptions, we are able to reduce the amount of features to be computed in next steps.

3. **PAA configuration.** The algorithm used to summarize the number of frames of each sub-activity sample was PAA. In this method,

the time series are divided into k segments of equal length and then each segment is replaced with a constant value, which is the average value of the segment. Then these average values are grouped in a vector, which represents the signature of the segment. In our experimentation, we have tested three different compression rates to compare which one performs better. The rates chosen were 2, 4, and 6, which reduce the frame samples to half, one quarter, and one sixth of the total size respectively. The best results were obtained with $PAA = 2$, and thus, this has been the configuration in the final experiments. If $X = \{x_1, .., x_n\}$ represents the time series, the PAA algorithm with compression rate K is represented by Equation 6.1, where $Y = \{y_1, .., y_m\}$ represents the resulting time series (i.e., y_i is the average of the K values x_j in the i-slot).

$$y_i = \frac{\sum_{j=K*(i-1)+1}^{K*i} x_j}{K}; \forall i \in \{1..\lfloor n/K \rfloor\} \tag{6.1}$$

Description	**Count**
Skeleton Features	**4**
- Left and right arm. Joint angle shoulder (joints *elbow-shoulder* & *shoulder-hip*)	2
- Left and right arm. Joint angle elbow (joints *shoulder-hand* & *elbow-shoulder*)	2
Objects Features	**16**
- Shortest distance to hand, group by object type (10 objects type)	10
- Shortest distance to hand, group by object id (maximum number of same objects type: 5)	5
- Sum of objects distances	1

Table 6.1: Features vector summary used in DTW algorithm.

4. **DTW configuration and recognition algorithm.**

After the execution of the previous steps, we have a final feature vector of 20 dimensions in each component, as shown in Table 6.1. One vector represents a set of frames of the video stream. Since the features vector are of different nature, they are normalized to the range [0,1] so that their weight does not unbalance the DTW algorithm operation. An in-depth description of the DTW algorithm may be found in [57].

The system training comprises the creation of a set of instances labelled with their corresponding sub-activity. Secondly, the DTW algorithm is applied for each pair of instances of the same activity in order to calculate the average distances between all of the training instances (*avgGlobal*, see Algorithm 1), that represents the average variation between different instances that the system may encounter during the recognition. The procedure $DTW(T_i, T_j)$ in Algorithm 1 returns the cost of the optimal path that matches T_i and T_j [57], and this value is normalized using the sum of the lengths of the instances being compared, to avoid that this value depends on the lengths of the instances. This process is done separately for each sub-activity to be trained.

```
Input: A finite set $T = \{T_1, T_2, \ldots, T_n\}$ of training instances for the
       same sub-activity. Each instance is a sequence of features
       vector (frames). $L_i$ is the length of each instance $T_i$
Data: The $DTW()$ function takes two sequences $T_i$, $T_j$ as input and
      calculates an optimal match of both, returning the calculated
      distance for them.
Result: avgGlobal is the average distance obtained after executing
        DTW for every pair of the training instances.
$avgGlobal \leftarrow 0$
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $n$ do
        $distance \leftarrow DTW(T_i, T_j)/(L_i + L_j)$
        $avgGlobal \leftarrow avgGlobal + distance$
    end
end
$avgGlobal \leftarrow avgGlobal/n * n$
```

Algorithm 1: Algorithm for sub-activities training using DTW.

Once the system is trained, the recognition of a new sample can be carried out following Algorithm 2. This algorithm is applied for each sub-activity, and returns an output score that measures the similarity between the new sub-activity acquired from the sensor data and the templates in the training set. As shown in Algorithm 2, we calculate the minimum distance between the new activity being recognized and the instances of each activity in the training set (*minDistance*), together with the the average distance to the training instances of each activity (*average*). Finally, a *score* in the range [0,1] (Equation 6.2) is calculated to obtain the similarity between the sample and the activity training templates. This score is composed of two parts: Firstly, the distance between the new sample and the nearest template in the training set for each activity, normalized to [0,1]. Secondly, the difference between the average distance between training templates calculated in Algorithm 1, and the average distance between the new sample and the training set. The higher the value of the score is, the more similar is the sample to the compared activity templates. The parameter α is used to state the relative relevance that should be given to the minimum distance and the difference of average distances during the recognition stage. In our experiments, we found that the value $\alpha = 1/7$ provided us with the best experimental results.

$$score \leftarrow$$

$$max\{NormalizeDistance(minDistance) - \alpha|avgGlobal - average|, 0\} \tag{6.2}$$

The sub-activity algorithm result with highest score will be the sub-activity selected as *recognized* for the input sequence. However, all the scores calculated for all the activities are transmitted to the high-level activity recognition module composed of the fuzzy ontology, in order to have complete information for complex activity reasoning and inference.

```
Input: The $newSequence$ is the sequence to be recognized.
       $L_{newSequence}$ is the length of $newSequence$
Data: A finite set $T = \{T_1, T_2, \ldots, T_n\}$ of training instances for the
      same activity. Each instance is a sequence of features vector
      (frames). $avgGlobal$ is the average distance obtained after the
      training stage. $L_i$ is the length of each instance $T_i$. The
      function $NormalizeDistance(..)$ normalizes the input distance
      to a value between 0 and 1.
Result: A $score$ for the $newSequence$, is a float value between 0 and
        1, being 1 the maximum score, related to the training set.
$average \leftarrow 0$
$minDistance \leftarrow \infty$
for $i \leftarrow 1$ to $n$ do
    $distance \leftarrow DTW(T_i, newSequence)/(L_{newSequence} + L_i)$
    if $distance < minDistance$ then
        $minDistance \leftarrow distance$
    end
    $average \leftarrow average + distance$
end
$average \leftarrow average/n$
$score \leftarrow$
$max\{NormalizeDistance(minDistance) - \alpha|avgGlobal - average|, 0\}$
```

Algorithm 2: Algorithm for sub-activities recognition using DTW.

6.2 Phase 2: Knowledge-driven context-aware modelling and recognition of high-level ADLs

In this section, we use ontological design principles to represent human activities semantically, and uncertainty and vagueness in the representation of information. The semantic inference-based module is based on fuzzy ontological rules [61] and takes as input the sub-activities detected in the first stage and their score, in order to detect high-level activities. Sub-activity scores are considered as a degree of certainty of activity detection, which is used to provide a reliable prediction and ontological reasoning considering uncertainty.

Ontological knowledge-based algorithm for high-level activities

The high-level activity recognition algorithm consists of a series of preprocessing steps before applying ontological reasoning itself. Three basic steps can be distinguished in the knowledge-based algorithm. They cover different aspects from the activity recognition process:

First, a high-level activity is composed of a sequential execution of a series of sub-activities, where each of these may use some object(s). When recognizing the activities, some of the sub-activities that compose it may be of more relevance within the performance of that activity than others. For instance, some sub-activities may be optional, and some users may not perform some of the most common sub-activities. Therefore, it is necessary to learn and compute the weights regarding the importance of each sub-activity in the recognition of each high-level activity. This is done in step 1.

In a second step, these sub-activities weights will be used in order to create specific rules that represent knowledge in the *fuzzyDL* KB. Finally, a heuristic engine is necessary to recognize the activities and deal with issues such as: a) recognizing activities of different duration (measured in number of sub-activities); b) sequences of sub-activities patterns that may be contained in the definition of more than one activity; c) activities that have very characteristic object usage, which makes them easy to discriminate from others. As some activities use almost the same objects, a same subset, and/or same sub-activities, the heuristic engine is crucial to distinguish among activities. Therefore, the engine was designed to discriminate and disclose each activity from the rest, according to a set of ordered filters and considering as much evidence as possible from sub-activities and object interactions. This evidence is used, in some filters, in form of manually codified sequences of sub-activities that characterize an activity. The heuristic engine is implemented in form of a pipeline, where each step in the pipeline is a filter that selects those activities that are most probable to be happening, according to the evidence of discrete events. One of the filters in the middle of the pipeline is invoking *fuzzyDL* inference reasoner to obtain the certainty (or degree of truth) of activities being happening. The rest of filters will help discern about the activity that is to be recognized.

Each discrete event is input to the ontological algorithm. At the same time, this input is an output from the data-driven sub-activity recognition module (Algorithm 2). Each atomic event is composed of a pair of a sub-activity and the objects used while it was performed. The detailed steps to recognize high-level activities are described next.

- **Step 1: Learn the weights for each sub-activity within an activity**

  Common sense knowledge is used to define domain knowledge, in our case, to define the home activities and the sub-activities that compose each of them. As we perform cross-validation to evaluate our rules on previously unseen users, we compute the value of the sub-activity weights for each sub-activity within an activity, based on the training dataset for certain users, and then test with the new unseen user.

  Sub-activity weights emphasize the importance of each sub-activity within the execution of an activity. Weights are computed based on a naïve approach in a previous configuration phase, on a semi-supervised manner, accounting for each sub-activity appearance within the activity performed in the labelled dataset. In other words, sub-activity weights represent the percentage or ratio of how much a sub-activity participates in the execution of a given activity. The average weight is computed and normalized [0,1] considering all sub-activities equally important, e.g., the weight of sub-activity a_j within the activity A_i is computed as in Equation 6.3:

  $$\frac{\#a_j}{\sum\limits_{i=1}^{n} \#a_i} \tag{6.3}$$

  where $\#a_j$ is the number of occurrences of sub-activity a_j within activity A_i, and n represents the amount of different sub-activities that participate in the given (high-level) activity A_i.

  Object interactions are associated to each sub-activity, and similarly, all objects initially have the same importance within the activity. Therefore, in the activity rule definition, all possible objects associated to a sub-activity within a given activity, are OR-ed. Examples of rules with different objects can be seen in Tables 7.21 and 7.22. For instance, this OR-relation is used when defining the sub-activity *reachMicroOrCloth*. The following *fuzzyDL* expression indicates that the pair composed of the sub-activity *reaching* and the used objects *microwave* or *cloth*, is defined by any user which performs the sub-activity *reaching*, while using, at the same time, either a *microwave* or a *cloth.*:

(define-concept reachMicroOrCloth (g-and User (some performsSubActivity (g-and reaching (some usesObject (or microwave cloth))))))

The usage of the OR relation in the definition of a sub-activity-object pair, such as in the definition of *reachMicroOrCloth*, is used if there is no unifying object category for all potential objects allowed to be used in that especific sub-activity. Let us put an example where such object category, that hierarchically unifies the use of several kinds of objects, is used. For instance, the sub-activity-object pair *moveDrinkingKitchenware*, is defined as any user which performs the sub-activity *moving*, while using any kind of object that inherits from the class *drinkingKitchenware*:

(define-concept moveDrinkingKitchenware (g-and User (some performsSubActivity (g-and moving (some usesObject drinkingKitchenware)))))

Tables 7.18 and 7.22 show how object categories were defined to group possible types of objects to be used within an adaptable activity definition, in a semantically logical way. Object categories allow to consider different objects usage, e.g., variations from person to person, or just different object usage associated to a given sub-activity. Categories were defined by observing the dataset, using common-sense knowledge, and creating a class for each category in the ontology, that inherits from the class *object*. For instance, we define in the fuzzy ontology the *drinkingKitchenware* class as a type (subclass) of *object*. Defined objects, inheriting from this class, are *bowl* and *plate*. Thus, all objects that represent *drinkingKitchenware* are defined as inheriting classes of this class. Let us see in *fuzzyDL* syntax how this is defined:

(define-concept bowl (and kitchenware stackable movable drinkingKitchenware containerKitchenware))

(define-concept cup (and kitchenware movable drinkingKitchenware containerKitchenware))

The lines above define the object *bowl* as an object of type *kitchenware, stackable, movable drinkingKitchenware,* and *containerKitchenware.* Analogically, the *cup* object is defined with multiple inheritance. In this case, the object category *drinkingKitchenware* is defined semantically as a type of *object*:
(define-primitive-concept drinkingKitchenware Object).

If new objects are integrated in the environment, for the system to keep working, only a new definition (as above, for *bowl*), would be necessary. This would make explicit the object categories to which the object belongs. Likewise, an associated weight for the sub-activity-object(s) pair, would need to be added to the ontology rules, for the activities affected.

- **Step 2: Create rules and represent fuzzy knowledge in the knowledge base**

  Activities are characterized by a set of previously configured sub-activities, in certain sequential order, that are required for the activity to be recognized. Each activity is recognized through a rule that contains a set of axioms involving sub-activities and the objects that these sub-activities use[2]. Because OWL (Web Ontology Language) does not allow order comparisons among two data properties' literal values in order to consider order among the sub-activities' timestamps, the heuristic engine filters take care of these sequential patterns, previously to executing the reasoner inference engine.

  In this phase we use the learnt weights from step 1 and each common sense rule to represent an activity. Rules are defined using ontological knowledge representation, which is fed to the KB in *fuzzyDL* reasoner. The syntax of rules, and the formal specification of concepts and relations for each activity, sub-activity, user and object in *fuzzyDL* can be seen in Tables 7.19 and 7.20.

- **Step 3: Run the heuristic engine to recognize an activity**

  Once defined the KB and rules, ontological inference can take place each time there is a change in the KB, e.g., a new sub-activity occurs, or every certain time interval. Semantic Web based reasoning is known to be powerful but computationally expensise. However, due

[2] These rules would be defined by an expert in classical expert systems; in our case, we use natural language and common sense descriptions for each activity, as well as observations from the dataset.

to recent advancements on the state-of-the-art reasoners, description logics have become more accessible and allow queries' responses to happen in seconds [158].

As we aim at recognizing critical activities online, i.e., in real-time, assuming a potentially high and rich traffic of multimedia events, we pre-filter some activities which are more prone to happen, based on some heuristics such as the probability of the activity to happen (based on the labelled occurrences of sub-activities). This is an example of first heuristic, that together with the following ones, optimize reasoning and recognition tasks using rich sensor streams. We will next describe each heuristic filter and how it selects or discards activities prone to be detected. If a filter does not select any candidate activity, the candidate activities, from the previous filter applied, are selected. This allows for propagation of the heuristics applied, so that next filter can be executed, taking as input the previous filter's output candidate activities.

The **pre-filter ratio** computes the proportion of sub-activities occurred in the most recent time window, from those required by the activity being evaluated. With "most recent time window" we refer to the most recently time window occurred in the discrete event stream-based dataset. This means we query for all possible activities that can have happened during the most recently occurred timed events (the last set of sub-activities with a start and end timestamps). The pre-filter selects those activities whose pre-ratio value is over a minimum threshold value. This threshold in [0, 1] is computed manually in an empirical form.

This heuristic considers the activities with highest proportion of unordered pairs of sub-activities and their object interactions associated. For instance, for the activity *making cereal*, the set of sub-activities (sub-activity-object(s) pairs) that compose it are:
moveMilkOrBowlOrBox, placeMilkOrBowlOrBox, reachMilkOrBowlOrBox, openMilkOrBox, reachMilkOrBowlOrBox, moveMilkOrBowlOrBox, pourMilkOrBox, moveMilkOrBowlOrBox, placeMilkOrBowlOrBox, reachMilkOrBowlOrBox, moveMilkOrBowlOrBox, pourMilkOrBox, moveMilkOrBowlOrBox, placeMilkOrBowlOrBox.

Activities filtered in pre-filter ratio are fed, in pipeline, to **filter 1**. Filter 1 heuristic considers, similarly to the pre-filter, the activities with highest proportion of ordered pairs of sub-activities and their object interactions associated. In this case, filter 1 assumes an ideal given order.

Filter 1's output activities (in its default, pre-filter ratio's output activities) are applied **filter 2**, which considers manually pre-defined order restrictions among (possibly repetitions of) subsets of pairs of sub-activities and objects used, and then keeps as candidate those activities with the highest ratio of these subsets. The specification of the order restrictions among subsets of pairs containing sub-activities and objects was done ad-hoc, using common sense but mainly by observing the dataset to perceive the sub-activities realized as well as the used objects in each activity. Since CAD-120 dataset authors did not provide a more detailed specification in natural language for each of the activity instructions given to the users that were recorded performing each activity, a "manual" observation of the dataset's employed objects was required. We can note that in [128] only a description for one activity was given and the rest was assumed to be self-explanatory by the high-level activity name.

In order to represent semantically order constraints, optionality of subsets of sub-activities-objects, and number of repetitions of these, we define the concept *activity subsequence*. Each activity subsequence is defined through the following triple: *(list of sub-activity-object pairs, optional, minNOfReps)*. The first element indicates a list of sub-activities-object pairs, the second element is a property that indicates if the whole activity subsequence is optional to recognize that activity pattern, and the third element represents the minimum amount required of repetitions of that activity subsequence, to be found in the activity stream. For instance, the pre-defined order restrictions among activities subsequences, for the activity *eating meal*, are defined as follows:

(reachCup, true, 1) (moveCup, eatCup, moveCup, false, 2) (nullSA, true, 1) (reachCup, true, 1) (moveCup, drinkCup, moveCup, false, 2) (placeCup, true, 1)

After the heuristic pre-selection filters are applied, the amount of queries to the reasoner is reduced. *FuzzyDL* is queried for the certainty of a set of activities happening, i.e., only those candidate activities which are the output of pre-filter, filter 1 and 2. Querying *fuzzyDL* is named in the algorithm as **filter 3**.

Next, the **filter 4** heuristic takes the highest certainty activity considering subsumption properties and concrete restrictions among the used objects' relative positions. The object positions filter is only used in case there are more than one candidate activity characterized by the same set of sub-activities and objects. For instance, *stacking* and *unstacking objects* are examples of such condition. The activity *stacking* is detected after finding at least 3 objects of the same type in unstacked position, and after that, the same type of objects in stacked position. The activity *unstacking* objects is detected when the same conditions, in opposite order, occur.

The dataset is very challenging since different objects can be used within the same activity (e.g. bowls, plates or pizza boxes for eating meal), and the same objects can be used for different activities (e.g. microwave is used for takeout food or cleaning). Stacking and unstacking activities were performed with pizza boxes, plates and bowls. Furthermore, the order of actions are not always the same (sometimes one takes the water glass before the pill when taking medicine, or the other way around) and some sub-activities are optional in cases (e.g., in making cereal, milk and cereal boxes do not always need to be opened, only when there is a new package). Also, when eating a meal, not always the person drinks something at the same time. There may also be other objects the user interacts with in between, and thus, it is not easy to model non deterministic behaviour. Furthermore, e.g., in the activity eating meal, the sub-activity of drinking and eating may be repeated an undefined number of times, for each user and time of the day. There are right and left-handed users, and in general, activities are not always performed in the same way. As some objects participate in different activities, it often happens that several activities have the same or very similar certainty of being happening. In that case, we select, if any, the one that is a *super-activity* or subsumes the other, regarding the sub-activity-object pairs that the activity definition contains. If no activity subsumes other, then the activity which subsumes the other regarding the longest duration, in number of sub-activities, is finally recognized. In order to disclose if there is an activity A_i that subsumes another A_j, we define the binary relation $ContainedIn(A_i, A_j)$ as a property that holds when all sub-activities in activity A_i appear within A_j's activity definition. E.g. in CAD-120 dataset, this property holds for *ContainedIn(takeout, microwaving)* and *ContainedIn(bending, arrangingObjects).* This property helps discriminating very similar activities that otherwise would often be recognized wrongly, e.g., *microwaving* and *arranging objects*, which are usually recognized as *takeout food.*

Summarizing, if there is a draw, or the activities with the two largest certainty values differ in a small difference , the following heuristic within filter 4 is applied:
(a) Select the activity that subsumes other activity in its specification, by sub-activity (SA), i.e., by having exactly the same SAs and object pairs. If not found, (b) is applied.
(b) Select the activity that subsumes others by cardinal of SAs (the activity's length). Assuming no concurrent activity happens and no

activity subsumes other (case (a)), check if from the candidate activities, there is one which has specification of longer cardinal of SAs than the rest. This favours the recognition of more complex activities if several activities of diverse length are final candidates. If no activity is found, (c) is applied.
(c) Select the activity with highest overall ratio, which is computed as an equally ponderated ratio of previously applied filter ratios (preratio, filter 1 and filter 2).

The AR algorithm described in Algorithm 3 summarizes all heuristic filters described. Each input data takes the following values:
[*Sub-activity, sub-activity ID, video start and end frames, current frame, object ID, object name, left and right hand distances to the object and objectPosition (X, Y, Z axis)*].

6.3 Semantic treatment of uncertainty, vagueness and incompleteness in context-aware activity recognition

As human nature is not deterministic and humans perform complex behaviours, uncertainty, incompleteness and vagueness are unavoidable aspects to consider when modelling human activities. These are inherent components to the human behaviour and need to be accounted for. Next, we detail several uncertainty sources and how is the modelling process in the fuzzy ontology, regarding the dataset used.

6.3.1 Incompleteness in activity recognition

Missing or failing sensor readings is one of the most typical sources of uncertainty in AR [209]. A concrete Ambient Assisted Living (AAL) experiment in [209] showed that 51 % of the system crash reasons were due to sensors out of battery, 22 % due to packet lost, 12 % due to a reasoning failure, 8 % due to sensors removed, and 7 % because of WiFi disconnection. By using a fuzzy ontology, a missing sensor reading does not drastically affect the recognition of a pattern, but rather diminishes the certainty of satisfiability of a given activity to be recognized as being happening. This is thanks to axiom definitions based on weighted concepts, which express the importance of a sub-activity associated to an object interaction. Fuzzy logic allow for flexibility or looseness in the model by accounting for the rest of components, if one in the definition is not instantiated.

```
Input: A sequence of sub-activities (associated with objects used)
       and their detection certainty (from Algorithm 2), in form of
       data stream
Data: A Knowledge Base $KB$ is given with a set of users, activities,
      sub-activities and object definitions. A set of rules
      $R = \{R_1, R_2, \ldots, R_n\}$, one per activity, is codified. Rules
      contain average weights for each associated sub-activity and
      object pair.
A time window $TW$ is provided. Its duration size is proportional to
each activity that is being considered as candidate to be recognized.
$ActivityRecognitionThreshold$: mininum activity recognition certainty
threshold to recognize an activity, given fuzzyDL reasoner's answer.
Result: $detectedActivity$: the recognized activity for the input TW
        (consisting of sequences of sub-activity-object pairs).
PRE-FILTER RATIO: Unordered sequence of sub-activity-object
pairs
$detectedActivities \leftarrow$
$filterActivitiesContainingSubActivityObjectPairConstraints(TW)$

RATIO 1 FILTER: Ordered sequences of sub-activity-object pairs
$detectedActivities1 \leftarrow$
$filterActivitiesFulfillingOrderAndObjectConstraints(detectedActi-vities)$
RATIO 2 FILTER: Activity sub-sequences (of sub-activity-object
pairs)
$detectedActivities2 \leftarrow$
$filterActivitiesFulfillingOrderAndObjectSubSequenceConstraints$
$(detectedActivities1)$
RATIO 3 FILTER: Find degree of certainty ($fuzzyDL$
min-satisfiability degree query) for each candidate activity
for $ActivityA_i$ in $detectedActivities2$ do
    $addgetCertaintyOfActivHappening(A_i)toactivitiesCertainties$
    if $getCertaintyOfActivityHappening(A_i)$
    $>ActivityRecognitionThreshold$ then
        add $A_i$ to $candidateActivities$
    end
end
if $candidateActivities.size() > 0$ then
    FILTER 4: Find relative object position-based activities,
    subsumed activities (and overall rate filter in case of certainty
    draw among several activities)
    $detectedActivity \leftarrow$
    $getActivityWithHighestCertaintyOverThreshold(activitiesCer-tainties, candidateActivities, ActivityRecognitionThreshold)$
end
return $detectedActivity$
```

Algorithm 3: Semantic high-level activity recognition

6.3.2 Uncertainty in sensor data acquisition

Pipeline-based or multi-level activity recognition has the disadvantage of incurring in error propagation. However, fuzzy ontologies allow declaration of axioms with a given truth degree for each atomic element (e.g., each sub-activity detected or each recognized object). Given the 3D-depth sensor certainty to recognize a sub-activity, a sub-activity instance can be input to the knowledge base to indicate its recognition with a certainty degree in [0, 1]. For instance, in *fuzzyDL*, stating that a sub-activity instance of type *placing* is detected with a degree of truth of 0.5, is defined as follows:
e.g.: *(instance placing subActivity 0.5).*

6.3.3 Vagueness in the importance of each sub-activity within a high-level activity

Not every user performs an activity on the same way. Some users change the predefined order in which they perform each sub-activity, and other users may skip some sub-activities depending on the context, or use different objects depending on preferences or situations (e.g., while eating, there is not a fixed predetermined number of repetitions for the sequence related to bringing the cutlery close to the mouth). These uncertainty aspects leave us room for abstraction when representing knowledge. We base our model or activity pattern on common sense lnowledge and observations from the dataset. Even when modelling these uncertain criteria, the semantic model should, in any case, maximize the degree of satisfiability or similarity to the defined fuzzy concept definition of activity. As indicated earlier, weights associated to the importance of each sub-activity within an activity definition, for each cross validation fold in our experiment, were taken from the dataset. However, if no evidence would exist, it is possible for the domain expert to set them ad-hoc.

6.3.4 Other vagueness and uncertainty sources in AR

Identifying the right user performing an activity is crucial to detect critical activities, as well as distinguishing among possible activities being performed concurrently. In multi-user scenarios 3D-depth sensors are expected to achieve very significant improvements in the very near future, and to reduce noise, e.g., in face or body recognition. These are other kind of uncertainty to be dealt with in the data acquisition phase. In our fuzzy ontology, we can state the certainty degree with which a user is identified, e.g., in *fuzzyDL*, *(instance Natalia User 0.9)* means that Natalia is an instance of the class User with a degree of truth of 0.9. We can also express the certainty with which the system identifies or recognizes a concrete user performing an activity. For instance, in *fuzzyDL*, *(related Natalia travelling performsActivity 0.9)* means that Natalia performs the activity travelling with a certainty degree of 0.9. These are just two examples on how any possible axiom can be upgraded by including an uncertainty degree *dimension.*

Detecting object interaction is another key context-aware component to discriminate among activities. However, the proximity of the user to objects does not always imply interaction. The closeness from the user's hands to the objects, as well as the relative distance among objects, are key to distinguish among activities which use the same (sub)sequences of sub-activities and the same kind of objects (e.g., in CAD-120, *stacking* and *unstacking objects*). Therefore, *DistanceToHands* and *maxDistanceAlongYAxis* are samples of thresholds used programmatically to deal with measurement and error variations. Likewise, the time window needs to adapt its size to a threshold-based buffer when querying for certain activity. In our case, we used a threshold summed to the max. execution time of a given activity. However, a fuzzy temporal window to express the times of the day when an activity can happen, can as well be utilized.

Chapter 7

Experiments and framework evaluation

> You need chaos in your soul to give birth to a dancing star
>
> Friedrich Nietzsche

In order to assess the suitability of our approach, there are several aspects to evaluate. First, we need to evaluate classical crisp ontology approaches in contrast to fuzzy ones, accounting for expressibility and performance with respect to the extra, but necessary computational resources. In this way we will assess the suitability of the proposed fuzzy ontology in Chapter 4. On the other hand, once evaluated the fuzzy ontology, we evaluate the overall hybrid (data and knowledge-driven) approach with regards to similar approaches in activity recognition. For this purpose we use a real public dataset. Next sections detail both experiments.

7.1 Experiment 1: Crisp vs fuzzy ontological reasoning for human behaviour recognition

In this section, we show the benefits of a fuzzy ontology for human behaviour recognition with respect to crisp approaches. With that purpose, we define two evaluation parameters. The scalability is understood as the capability of the ontology to perform with a rule set and a reasoner to achieve activity recognition, in reasonable execution time, for large amounts of data size (KB' size). The satisfiability degree (or firing accuracy of rules, $\in [0,1]$) is another parameter considered, not directly present in crisp ontologies, where either an activity is recognized or not. The satisfiability degree influences the recognition accuracy and gives more precise information about the certainty of having recognized certain activity happening.

Rule 1 description:	If a person opens the kitchen cupboard to take coffee, turns the coffee machine on, holds the coffee jar, and possibly takes milk from the fridge, then the person is making coffee.
Definition:	*(define-concept antecedent1 (g-and User (some performsAction MakeCoffee)))* *(define-concept MakeCoffee (w-sum (0.1 OpenCupboard) (0.5 TurnCoffeeMachineOn) (0.3 MoveCoffeeJar) (0.1 OpenFridge)))* *(define-concept consequent1 (g-and User (some performsActivity MakingCoffee)))* *(define-concept Rule1 (l-implies antecedent1 consequent1))*
Rule 2 description:	If a meeting room has the lights on, there is a meeting going on at the room.
Definition:	*(define-concept antecedent2 (g-and MeetingRoom (some hasLights (some hasStatus ON))))* *(define-concept consequent2 (g-and MeetingRoom (some hasOcupancyStatus HoldingAMeeting)))* *(define-concept Rule2 (l-implies antecedent2 consequent2))*
Rule 3 description:	If a person makes use of a bottle, a plate, a fork, a spoon, and a knife, we have high certainty that he/she is having lunch.
Definition:	*(define-concept antecedent3 (g-and User (some performsAction Lunch)))* *(define-concept Lunch (w-sum (0.1 UseBottle) (0.4 UsePlate) (0.2 UseFork) (0.2 UseSpoon) (0.1 UseKnife)))* *(define-concept consequent3 (g-and User (some performsActivity HaveLunch)))* *(define-concept Rule3 (l-implies antecedent3 consequent3))*
Rule 4 description:	If a person goes out of his/her office, passes by the corridor, opens the storage room, and takes the exercise stick, then he/she is performing a Keppijumpa stretching exercise.
Definition:	*(define-concept antecedent4 (g-and User (some performsAction KeppijumpaStretchingExercise)))* *(define-concept KeppijumpaStretchingExercise (w-sum (0.15 ExitOffice)(0.15 ExitCorridor) (0.15 OpenStorageRoom) (0.55 UseStick)))* *(define-concept consequent4 (g-and User (some performsActivity DoStretching)))* *(define-concept Rule4 (l-implies antecedent4 consequent4))*

Table 7.1: Fuzzy Rules 1-4.

Rule 5 description:	If a user's phone is in a location near the office of the head of the laboratory, then the user is having a meeting with his/her supervisor.
Definition:	_(define-concept antecedent5 (g-and User (some hasPhone (some isInLocation (some isNearTo JohanLiliusOffice)))))_ _(define-concept consequent5 (g-and User (some performsActivity MeetingSupervisor)))_ _(define-concept Rule5 (l-implies antecedent5 consequent5))_
Properties for Rules 6, 7, 8:	Fuzzy properties for representing the personal average sleep quality in number of sleeping h/day, average number of steps walked per day, and overall immune-defense level for overall health in a integer scale of [0, 10].
Definition:	_(functional hasAvgSleepQuality)_ _(functional didAvgNSteps)_ _(functional hasImmuneDefenseLevel)_ _(range hasAvgSleepQuality *real* 0.0 20.0)_ _(range didAvgNSteps *integer* 0 50000)_ _(range hasImmuneDefenseLevel *integer* 0 10)_ _(define-fuzzy-concept lowImmuneDefenseLevel left-shoulder(0, 10, 0, 5))_ _(define-fuzzy-concept badSleepQuality left-shoulder(0.0, 20, 4.0, 5.0))_ _(define-fuzzy-concept highNSteps right-shoulder(0, 50000, 7000, 10000))_ _(define-fuzzy-concept lowNSteps left-shoulder(0, 50000, 2000, 2500))_
Rule 6 description:	If sleep quality is very bad and pedometer-based step counter measures a low number and stress is high, then immune-defense is low (having a coffee break may be recommended)
Definition:	_(define-concept antecedent6 (g-and (some hasAvgSleepQuality badSleepQuality) (some didAvgNSteps highNSteps)))_ _(define-concept consequent6 (g-and User (some hasImmuneDefenseLevel lowImmuneDefenseLevel)))_
Rule 7 description:	Analoge to Rule 6 but caused when the person is not having enough exercise
Definition:	_(define-concept antecedent7 (g-and (some hasAvgSleepQuality badSleepQuality) (some didAvgNSteps lowNSteps)))_ _(define-concept consequent7 (g-and User (some hasImmuneDefenseLevel lowImmuneDefenseLevel)))_
Rule 8 description:	A disjunction among two previously defined rules
Definition:	_(define-concept Rule8 (g-or Rule6 Rule7))_

Table 7.2: Fuzzy Rules 5-8.

Although the literature offers a wide variety of activity recognition datasets, it is rare to find them expressed in the form of semantic ontology-based axioms. One exception is the Opportunity Dataset [143], adapted to an ontological framework [96]. Their multilevel activity ontology-based dataset is validated with ELOG reasoner and it shows a high degree of concurrency in fine-grained activities [96]. ELOG is a probabilistic reasoner for OWL $\mathcal{EL}$, focused on the log-linear description logic based on $\mathcal{EL}++$ without nominals and concrete domains ($\mathcal{EL}$++-LL). It is worth noting that the computational cost of OWL $\mathcal{EL}$ is much lower than for *fuzzyDL*. However, this is a trade-off that makes *fuzzyDL* stronger due to its $\mathcal{SHIF}$ expressivity, much superior to the one of OWL $\mathcal{EL}$.

When dealing with uncertainty, it is important to distinguish when probabilistic reasoning is suitable with respect to fuzzy reasoning. Probabilistic reasoning can model uncertainty associated to previous experience cases. However, fuzzy reasoning can help modelling vagueness or natural language-based descriptions, based on knowledge representation. For instance, *almost every day* Peter has muesli for breakfast. Expressions such as "almost every day", "quite", "little", etc. could be modelled to better preserve natural language when expressing rules from experts. To validate our own ontology, we create a fuzzy rule KB for human activity recognition with concepts and relationships. In addition, for more complex queries, based on triple patterns such as in standard SPARQL queries, a mapping can be seen in [66] between triple pattern queries (s, p, o) and *fuzzyDL* queries. E.g. for queries with wildcards such as, e.g., (?, p, o), the following *fuzzyDL* query could be used to find the minimal degree of satisfiability for that given predicate form:
If D ∈ p.Domain: ∀ Individual i ∈ D: (min-related? i o p).

For these reasons, as we require a fuzzy reasoner due to semantics, we cannot reuse existing datasets and therefore, we designed our own set of rules for *fuzzyDL* 2.04 (using Gurobi optimizer 5.0.2).

The rules we used for the ontology validation experiment are shown in Tables 7.1 and 7.2, but the whole file with *fuzzyDL* instances and other definitions and queries is available online together with the ontology[1][HAR]. Additional information about the *fuzzyDL* syntax may be found in `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html`.

The goal of our experiment is to verify that fuzzy ontologies can outperform crisp ontologies in human behaviour recognition. Thus, for the test, we created 8 implicative rules that were applied over the fuzzy approach and its corresponding crisp ontology. Rules 1 to 5 test flexibility in the sense that skipping some actions does not prevent the recognition of a behaviour.

[1] Fuzzy Human Behaviour Ontology, rules and queries: `http://users.abo.fi/ndiaz/public/FuzzyHumanBehaviourOntology/`

Description	*fuzzyDL* Code
Query 1	*(min-instance? Victor (some performsActivity MakingCoffee))*
Query 2	*(min-instance? ADARoom (some hasOcupancyStatus HoldingAMeeting))*
Query 3	*(min-instance? Robin (some performsActivity HaveLunch))*
Query 4	*(min-instance? Natalia (some performsActivity DoStretching))*
Query 5	*(min-instance? Ana (some performsActivity MeetingSupervisor))*
Query 6	*(min-instance? facts Rule6)*
Query 7	*(min-instance? facts Rule7)*
Query 8	*(define-concept Rule8 (g-or Rule6 Rule7))* *(min-instance? facts Rule8)*
Facts for queries 6, 7 and 8	*(instance facts (= hasAvgSleepQuality 3) 0.7)* *(instance facts (= didAvgNSteps 19000))* *(instance facts (= hasAvgSleepQuality 4) 0.2)* *(instance facts (= didAvgNSteps 2000) 0.9)*

Table 7.3: Fuzzy Queries.

These rules were designed with the aim of giving different weights to certain actions within an activity, so that weights can strengthen the importance of the different actions that compose an activity to finally give an overall degree of satisfiability. On the other hand, rules 6, 7, and 8 use fuzzy membership functions such as the following:

(define-fuzzy-concept lowImmuneDefenseLevelCrisp left-shoulder(0, 10, 0, 5))

(define-fuzzy-concept badSleepQualityCrisp left-shoulder(0.0, 20, 4.0, 5.0))

(define-fuzzy-concept highNStepsCrisp right-shoulder(0, 50000, 7000, 10000))

(define-fuzzy-concept lowNSteps left-shoulder(0, 50000, 2000, 2500))

To transform these fuzzy concepts to the crisp case, thresholds need to be taken from the membership functions above and then hard-coded to work as thresholds as shown by the next lines:

(instance lowImmuneDefenseLevelCrisp ImmuneDefenseLevel)

(define-concept antecedent6 (and (<= hasAvgSleepQuality 5) (>= didAvgNSteps 7000)))

(define-concept consequent6 (and User (some hasImmuneDefenseLevel lowImmuneDefenseLevelCrisp)))

This rule formulation for the crisp case shows more limitations in expressivity and further reasoning than the fuzzy approach, since the formulation of the rule above would not allow linguistic variables for further imprecise queries of the type *(all-instances? (some hasImmuneDefenseLevel lowImmuneDefenseLevel))*.

After creating the set of rules, we instantiated individuals and formulated queries 1 to 8 in Table 7.3, to verify the triggering of rules 1 to 8, respectively. All queries were designed so that they should fire (in fuzzy case, to certain degree), except rule 6, that is made to not fire at all. On the fuzzy case, we assume that the firing of a rule to detect a behaviour is subject to an activity/behaviour-dependent threshold. For the sake of simplicity, in Table 7.4, we considered as fired those rules whose answer had a degree of satisfiability larger than zero.

This experiment helps us to verify the expressiveness power and practical implications of fuzzy ontologies with respect to crisp approaches. In Table 7.4, none of the rules fires except rules 7 and 8 for the crisp case. On the other hand, all rules except rule 6 fire in the fuzzy case, with a degree of truth or certainty. An example is rule 4: Here, *KeppijumpaStretchingExercise* is a concept that is the weighted sum of the concepts it is composed of. In this case, if the instance *Natalia* performs the actions going out of her office, exiting the corridor, opening the storage room and using the exercise stick, we can recognize that she did the stretching exercise session. This is expressed in *fuzzyDL* as follows:

(related Natalia exitOffice performsAction 0.6)
(related Natalia exitCorridor performsAction 1.0)
(related Natalia openStorageRoom performsAction 1.0)
(related Natalia useStick performsAction 1.0)

In the crisp case, if for any reason, one of these actions is not detected, the activity *DoStretching* is not recognized, as Table 7.4 shows. However, the fuzzy approach still fires the rule with a certainty degree value lower than 1. To implement this behaviour in the crisp ontology, we need to establish a threshold criteria for each fact, behaviour, or object to be considered in the case that the certainty of a fact is not 1, so that the rules can fire. This may be done manually or using optimization techniques such as genetic or evolutionary algorithms. This is part of future work directions on specific domain scenarios where threshold values need to be tuned and optimized accordingly. However, a fuzzy approach deals with uncertainty and eases the management of these situations. Thus, we may conclude that fuzzy ontologies can provide improvements in expressivity with respect to the crisp cases, but also in performance due to their power to manage uncertainty.

Query	**Crisp Answer**	**Fuzzy Answer**
Query 1	0	0.1
Query 2	0	0.5
Query 3	0	0.4
Query 4	0	0.4
Query 5	0	0.8
Query 6	0	0.0
Query 7	1	1.0
Query 8	1	1.0
Firing Accuracy	37.5%	100%

Table 7.4: Degree of satisfiability for crisp VS fuzzy (equivalent) human behaviour rules. In crisp case, 0 = No firing; 1 = Firing.

Having shown the benefits of fuzzy ontologies over crisp ones, in modelling human behaviour, we are also interested in evaluating if our approach scales to large sizes of Knowledge Bases. With this purpose, we measured execution times for queries 1 to 8. We implemented the equivalent set of crisp queries in *fuzzyDL* as well, to evaluate performance and recognition accuracy and to better assess the rules' scalability factor in practice. Both fuzzy and crisp experiments can be compared in Table 7.5, and the differences among average times of our 8-query dataset can be seen in Table 7.6. We must notice that execution time does not vary substantially for fuzzy queries with respect to crisp queries. However, a larger increase in average time occurs when we enlarge the KB dataset to at least 10^5 instances in both crisp and fuzzy cases. In the fuzzy case, query average response time goes to the order of up to $\backsim 13-16$ minutes. We are aware that this delay may be too long if the purpose is to notify about safety-critical activities. Therefore, in the future, we will focus on prioritizing the detection of critical activities so that the current reaction time obtained can decrease.

It is important to note that, although KBs of size 10^5 impose a considerable increase of execution time with respect to KBs of size 10^4 (from orders of about 300 times more), this is not significant due to the comparison with respect to crisp reasoning in KBs of the same size. In fact, importantly enough, for KB sizes of 10^5 instances, some queries are faster in the fuzzy case than in the crisp one. *FuzzyDL* internal reasoning tableau algorithm optimizations for large numbers of axioms/instances, Java virtual memory swapping or cache memory functioning seem to be implementation-related reasons for this phenomenon to happen. In any case, it is worth mentioning that all times were measured when running each query isolated and independently from others. This is to make queries comparable, since once a first query has been answered, the rest of queries take much less time due to the reuse of the internal graph model already built. As conclusion, we can affirm that fuzzy ontology-reasoning for activity recognition is scalable.

Both crisp and fuzzy experiments, measuring query running time in seconds, were run on an Intel(R) Core i7-4500@1.80 GHZ 2.40 GHZ, 8 GB RAM 64-bit and Windows 8.1. A limitation of *fuzzyDL* is the maximum number of individuals allowed (maxIndividuals = 1000000000). Nevertheless, a KB of size = 10^6 was not handled by Java memory.

After experimenting with a real life and complex enough ontology, we can affirm that there is a need for more complete fuzzy reasoners that can handle real-time notifications (such as a subscription mechanism as in e.g. M3 RDF store [102]) to avoid bottlenecks with constant querying. This does not occlude *fuzzyDL*'s potential and the fact that it has shown to be successful in diverse domains. In our presented case study, modelling rules in *fuzzyDL* also showed some challenges. As *fuzzyDL* does not allow yet to express implication rules where the subject in a triple (s,p,o) is a concrete individual, our experiment focused on general rules for individuals of a given class C that acts as the subject of the query. In the future, we expect to have more efficient ways of concreting the rule so that it can specifically apply to unique instances/individuals so as to achieve a proper rule personalization. At the moment, three workarounds solve this situation. a) An extra class (e.g. *NataliaClass* concept) can be created for each rule we want it to solely affect to a concrete individual (e.g. individual *Natalia* from class *User*). This makes explicit, by naming a class with an individual's name, that that rule should only apply to the given individual. b) The firing of the rule for a given individual can be detected by first instantiating the individual of interest (in this case *Natalia*) as an instance of that rule (e.g., Rule 4): *((instance Natalia Rule4)* . This step is required to give a correct answer in the second step. Secondly, a rule firing can be detected by querying the degree of satisfiability of individual *Natalia* satisfying Rule 4. This is done by querying for *Natalia* being an instance of the rule's consequent (this applies to the current state of the KB). E.g.: *(min-instance? Natalia (some performsActivity DoStretching)).* c) It is also possible to find all individuals satisfying a given rule by querying: *(all-instances? Rule4)* after having instantiated all individuals that we want the rule to apply to (as in step b). E.g.: *(instance Natalia Rule4).*

Despite some minimal workarounds required by *fuzzyDL*, we demonstrate that fuzzy ontologies may be more realistic and provide better accuracy for human behaviour recognition than crisp ontologies. Experiment 1 showed that if facts are not completely true, crisp rules cannot fire, while a fuzzy approach fires them with a satisfiability or certainty degree. Solving this situation in crisp solutions would require a continuous threshold management that would make the problem more complex, while fuzzy systems deal with this type of situations in a natural way. Apart from being more accurate, we also showed that the fuzzy approach is scalable for larger sizes of KBs.

To summarize, varied (from fine to coarse-grained) levels of abstraction were provided to identify atomic actions, activities composed of actions, and behaviours comprised by an aggregation of the latter two. Recursion, for more flexible and scalable modelling, is also allowed at activity and behaviour levels. A behaviour can be customized and associated to a unique user, user group, or certain type of action, activity, or context dependence. As a final remark, and considering future larger data size scenarios, we believe on the potential of combining the use of a crisp-fuzzy hybrid architecture approach of KBs for performance and scaling reasons. This would allow advantages of both paradigms to be fully exploited (see further discussion in [66]). The following sections proposals take this idea to a next level.

Crisp and Fuzzy Query	KB size =100	KB size =1000	KB size =10000	KB size =100000
Crisp Q1	0.0754	0.1688	3.2986	911.222
Crisp Q2	0.074	0.1656	3.0992	962.054
Crisp Q3	0.0752	0.1622	3.3778	984.197
Crisp Q4	0.0752	0.1606	3.2124	1071.07
Crisp Q5	0.0778	0.1656	3.3284	1076.05
Crisp Q6	0.0834	0.156	3.2374	1001.38
Crisp Q7	0.078	0.1656	3.0688	1021.18
Crisp Q8	0.078	0.1626	3.2016	1082.07
Fuzzy Q1	0.0846	0.172	3.1594	800.31
Fuzzy Q2	0.078	0.1626	3.2844	1008.21
Fuzzy Q3	0.0814	0.1626	3.5488	1025.97
Fuzzy Q4	0.078	0.1656	3.4594	912.07
Fuzzy Q5	0.0782	0.175	3.3816	1138.75
Fuzzy Q6	0.0812	0.175	3.5252	1128.29
Fuzzy Q7	0.0782	0.1812	3.3534	978.135
Fuzzy Q8	0.0876	0.1814	3.7248	1078.7138

Table 7.5: Average execution time (in s) for each rule (fuzzy and crisp) in different Knowledge Base (KB) sizes in number of individuals/instances (*User*).

7.2 Experiment 2: Hybrid data-driven and knowledge-based fuzzy ontological reasoning for real-time activity tracking and recognition

In this section, we extend the work in previous section [61] by not only using an external public dataset (Cornell Activity Dataset CAD-120) to validate our ontology-based framework, but we also plug a previous data-driven phase to recognize sub-activities, and allow flexibility and adaptability to changes in the way activities are performed. Further modelling of object interaction and activity recognition heuristics are also proposed and evaluated. In this

KB size in number of individuals (*User*s)	Average query running time (s) difference (fuzzy-crisp)	Standard deviation
10^2	0.0038	0.0047
10^3	0.0086	0.0086
10^4	0.2016	0.205
10^5	-4.8468	128.8485

Table 7.6: Average execution time differences (in s) for the 8 queries (fuzzy and crisp) dataset and different Knowledge Base (KB) sizes in number of individuals/instances (*User*).

way, it can be demonstrated that, by using a fuzzy ontology-based hybrid approach, the benefits of expressibility and looseness in the activity models allow not only for higher accuracy, but also for more tolerance to inherent uncertainty and vagueness.

7.2.1 CAD-120 3D-Depth dataset

Although the literature offers a wide variety of activity recognition datasets, it is hard to find one with a enough diversity to test fine and coarse-grained activities in RGB-D video, and where semantics features can be tested, together with object interaction, to allow discrimination of activities according to context. The dataset that best suits our requirements for different levels of activity recognition is the recent CAD-120 dataset (Cornell Activity Dataset) [128]. It is a very challenging dataset that contains 120 activities with 1191 sub-activities performed by 4 subjects: two male and two female (one of them left-handed). It contains the following high-level activities, sub-activities and labeled objects:

- 10 high-level activities: *making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, taking food, arranging objects, having a meal.*
- 10 sub-activity labels: *reaching, moving, pouring, eating, drinking, opening, placing, closing, scrubbing, null.*
- 10 objects: *book, bowl, box, cloth, cup, medicine box, microwave, milk, plate, remote.*

  The goal of our experiment is to test the performance of our approach under complex AR scenarios by adding semantics, through fuzzy ontology-based context reasoning, to data-driven AR. With that purpose, we define the parameters in Equations 7.1, 7.2, 7.3, where *tp* stands for true positives, *tn* are true negatives, *fp* is false positives and *fn* for false negatives.

$$precision = tp/(tp + fp) \tag{7.1}$$

$$recall = tp/(tp + fn) \tag{7.2}$$

$$accuracy = (tp + tn)/(tp + tn + fp + fn) \tag{7.3}$$

As a second evaluation metric, we are interested in the scalability of the system, as a reactive system. Scalability is understood as the capability of the ontology to perform with a rule set and a reasoner to achieve AR, in reasonable execution time, for large amounts of data size. The system showed to be scalable in this sense [61], but in this occasion we aim at having a whole hybrid AR system that can assist users, responding to changes or special situations in the environment, in real-time. For this reason, we use a metric that is critical for system responsiveness in order to assess the real-time system performance for continuous AR [49]: The Time per Recognition Operation (TpRO) is defined as the interval in second(s) from the time a sensor is activated until the time an activity is recognized. Results of our approach, for these metrics, in each of the two phases of the algorithm, are shown in the next subsections.

We performed leave-one-out cross-validation in each of the two phases as in Koppula et al. [128] (CAD-120 dataset's authors), using each subset of 3 users for training and tested with the fourth one, to be able to compare our approach with the previous method accurately.

7.2.2 Evaluation of the data-driven recognition of sub-activities

Table 7.7 shows the average results of the cross-validation process for sub-activity labelling.

We have as well measured the recognition execution time for each sample, and calculated the average. These experiments were done running

	Reaching	Moving	Pouring	Eating	Drinking	Opening	Placing	Closing	Scrubbing	null
Reaching	.94	.01				.01	.02	.01		.04
Moving	.02	.89	.02	.01	.01		.04			.02
Pouring		.02	.96				.02			
Eating	.02	.06		.84	.03					.05
Drinking	.02	.08			.81		.02			.08
Opening	.03	.02				.91		.02		.01
Placing	.02	.04	.01				.90			.02
Closing	.03	.01				.03	.01	.88	.01	.02
Scrubbing		.02				.02		.04	.92	
null	.02	.02			.01		.01			.93

Table 7.7: Confusion matrix for sub-activity labelling

the developed application over *Ubuntu 12.10* as *OS*, in a PC with a *Pentium Dual-Core processor (CPU E5700, 3.00GHz, 800 MHz FSB, 2GB RAM)*. Each sub-activity sample has different duration, varying from 10 to 510 frames. There were 11 sub-activity samples, out of 1191, in the CAD-120 dataset with duration lower than 10 frames (one third of a second), but we have discarded these ones as we think they are possibly due to some misprints when labelling the data, as their length greatly differs with the average of their kind. We present these results on Table 7.8. As it can be observed, the average sub-activity recognition time is 178.99 milliseconds, and since the average sub-activity duration is 50.8 frames, this means our recognition algorithm is able to process more than 380 frames in less than one second, in a medium range 5 years old CPU.

Sub-Activity Recognition Time	**Average Time**
Reaching	138.87
Moving	193.9
Pouring	279.55
Eating	141.33
Drinking	178.5
Opening	284.87
Placing	142.46
Closing	241.58
Scrubbing	532.99
null	173.02
Average	178.99

Table 7.8: Average recognition times (in milliseconds) per sub-activity.

Table 7.9 shows the results obtained for the experiment carried out. We consider the comparison with the basic method, where Koppula et al. [128] obtained 76.8% average accuracy, 72.9% precision and 70.5% recall (overall in average with ground truth temporal segmentation and object tracking). We observe an increment in the results with the solution we propose. This is due to the usage of our framework described in Section 6, which achieves 90.1% average accuracy according to the results shown previously. Thus, the approach presented in this work is highly competitive.

In order to verify these results, we applied a statistical analysis to evaluate if the improvement is statistically significant. The null hypothesis of equal performance between classifiers is rejected according to the sample t-test of Student for $\alpha = 0.05$ with a p-value of 7.3632e-04. As the hypothesis has a p-value ≤ 0.05, there is a statistically significant difference on improvement between Koppula et al. [128] and our recognition algorithms.

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
Koppula et al. [128]	76.8±0.9	72.9±1.2	70.5±3.0
Our Method	90.1±8.2	91.5±4.6	97.0±5.8

Table 7.9: Comparison of our approach with Koppula et al. [128] for the CAD-120 dataset sub-activity recognition. Average *Accuracy, Precision* and *Recall*.

7.2.3 Evaluation of the knowledge-based recognition of high-level activities

The main features of the CAD-120 dataset were adapted to ontological concepts and relations. In order to semantically define an activity, different concepts and relations must be explicitly defined, as well as the order relations among sub-activities. For this purpose, we use given high-level descriptions of the activities which were asked to users to be performed (multiple times with different objects). For example, the instructions for making cereal were: 1) Place bowl on table, 2) Pour cereal, 3) Pour milk. A summary of the rest of activities is described in Table 7.10.

Object interaction in the ontology was modelled by categorizing each object by its usage in order to discriminate among activities. These semantic categories or super classes are in Table 7.11, while the object and data properties modelled are in Table 7.12.

When it comes to populate the ontology with real data, instances of each class, called individuals, are created. Individuals are created in OWL

Activity	**Semantic concept**	**Description**
Making cereal	*cereal*	Take cereal box, bowl and milk (open them) and pour both.
Taking medicine	*medicine*	Take medicine box from cupboard, take glass, eat pill and drink water
Stacking objects	*stacking*	Stack on a pile plates, boxes or bowls
Unstacking objects	*unstacking*	Unstack from a pile plates, boxes or bowls
Microwaving food	*microwaving*	Take food container or kitchenware, place it into microwave and take it out
Picking objects (bending)	*bending*	Pick up an object from the floor
Cleaning objects	*cleaningObjects*	Clean up objects (microwave with a cloth)
Taking out food	*takeout*	Take food and heat in microwave
Arranging objects	*arrangingObjects*	Arranging on a table, e.g. setting up the table
Having meal	*eatingMeal*	Eating a meal on the table

Table 7.10: Semantic description of each high-level activity

through an instantiation in form of an RDF triple (subject, predicate, object). Ontological relations represent a property among two classes (called object properties) or among a class and a datatype (called data properties). Therefore, the content in Table 7.12 specifies the ontological schema in which instances (in form of triples) can be represented, conforming to the types specified by the subject (the domain of the relation), the predicate (the object/data property) and the object (the datatype range of the relation). Once provided these values, in a way that the ontology keeps being consistent, the reasoning engine can perform inference of new relations and properties automatically.

The rules created for the ontology validation experiment are shown in Tables 7.19 and 7.20, and the concepts, properties and axioms that the rules use are defined in Tables 7.12, 7.18, 7.21 and 7.22. Additional information about the *fuzzyDL* syntax may be found in *fuzzyDL* website[2].

We believe that this dataset is accurate and close to real scenarios because objects are used in different tasks with different purpose, and the same activity could be performed with different objects. For example, the stacking and unstacking activities were performed with pizza boxes, plates and

[2] *fuzzyDL*: `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html`

Object	**Semantic description based on object category super-concepts**
book	*(and arrangeable movable)*
bowl	*(and kitchenware stackable movable drinkingKitchenware containerKitchenware)*
box	*(and stackable movable pickable arrangeable)*
cloth	*(and kitchenware movable)*
cup	*(and kitchenware movable drinkingKitchenware containerKitchenware)*
medicineBox	*(and edible movable)*
microwave	*(Object)*
milk	*(and edible movable)*
plate	*(and kitchenware stackable movable containerKitchenware)*
remote	*(Object)*

Table 7.11: Objects' semantic description based on usage-driven object categories.

bowls. Activities were performed through a long sequence of sub-activities, which varied from subject to subject significantly in terms of length of the sub-activities. The order in which the sub-activities were executed within a task can also differ.

Confusion matrices in Tables 7.13 and 7.14 show in detail accuracy values for each activity with two settings: Firstly, in combination with the data-driven module that returns the detected activity, and secondly, replacing the data-driven module with the ground-truth sub-activities detection. These two settings help to verify the influence in the accuracy with respect to the low-level activity recognized, and also to know the limits of our fuzzy ontology approach.

Regarding the performed experiment, Table 7.16 shows the results obtained. We set as baseline the method proposed by Koppula et al. [128], using CAD-120 dataset, which results on a 79% accuracy, 78.6% of precision and 78.3% of recall overall in average. We appreciate an improvement in the knowledge-based solution we propose, increasing the average accuracy to 82.9%, and the precision and recall, respectively, to 84.1% and 97.4%.

In order to evaluate the effects (or costs) of the manual common-sense knowledge codification work, reflected on the usage of the heuristic in Algorithm 3, we performed a naïve implementation where no heuristic filters were applied. Instead, we solely applied *fuzzyDL* reasoner to find out the activity with highest certainty for it to be predicted[3]. Results are also shown

[3] In this setting, only the pre-filter ratio in Algorithm 3 was used, which provides information to query only for those activities for which any kind of evidence has occurred in the last event time window, i.e., we only query for activities for which any of their involved sub-activities has occurred.

Description	Object Properties	Domain	Range Property
High-level activity performed	*performsActivity*	User	Activity
Sub-activity performed	*performsSubActivity*	User	SubActivity
Object interaction within a sub-activity	*usesObject*	SubActivity	Object
Description	**Data Properties**	**Domain**	**Range Property**
Sub-activity start frame	*hasStartFrame*	SubActivity	integer functional
Sub-activity end frame	*hasEndFrame*	SubActivity	integer functional
Object position in X axis	*hasPosX*	Object	*double* 0 100000
Object position in Y axis	*hasPosY*	Object	*double* 0 100000
Object position in Z axis	*hasPosZ*	Object	*double* 0 100000

Table 7.12: Fuzzy Roles (Ontology Object and Data properties)

in Table 7.16 and the respective confusion matrix is in Table 7.15.

Avoiding the application of the heuristic filters provides poor results when compared with the whole hybrid system, and therefore, their absence makes the method incomplete. In this case, too high rate of false positives and negatives was obtained for the recognition of high-level activities. This is explained by the fact that DL reasoners are based on monotonic logic, which means that addition of new axioms into the KB can be performed, but no retraction of information takes effect. In other words, the certainty of an axiom can be increased but not decreased. The inability to retract facts within a time window of a given size, makes the manual pre-definition of sub-activity ”profile” filters necessary. These filters required common-sense knowledge and observation of the dataset to determine a set of subsequences of sub-activities and object patterns in each activity. In the heuristic filters algorithm, after querying *fuzzyDL* reasoner for each activity' satisfiability degree, the filters are applied to avoid too high false positive and negative rates, as well as for improving the accuracy, precision and recall metric values. Too high false positives would be even more noticeable in potential cases such as multi-users, parallel or interleaved AR settings (which is not the case for CAD-120). In any case, logic monotonicity is an inconvenience to deal with when using DLs, and future solutions should be seek for larger stream dataflows.

On the other hand, if we assume an ideal scenario with 100% accuracy on labelled input sub-activities, i.e., supposing all sub-activities are properly recognized in the first phase, then a precision of 90.8%, a recall of 98.1% and an accuracy of 91.07% are achieved. Both experiments in the sub-activity and high-level activity tracking and recognition (both in CAD-120 and our framework) were realized following the settings in Koppula et al. [128], i.e., assuming ground truth temporal segmentation is given.

Furthermore, we executed a statistical analysis in order to evaluate if our improvement is statistically significant. The null hypothesis of equal performance between classifiers is rejected according to the sample t-test of Student for $\alpha = 0.05$ with a p-value of 0.0013. As the hypothesis has a p-value ≤ 0.05, there is a statistically significant difference between Koppula et al. [128] and our recognition algorithms. In addition, our average standard deviation from all mean precision values is smaller than the one for Koppula et al. [128], as we have $\sigma = 0.0092$ while Koppula et al. [128] produce $\sigma = 4.1$ for the full model with tracking capabilities in the real life scenario (non-ideal case with predicted sub-activities as input). Final overall comparison statistics are shown in Table 7.16.

Regarding AR execution times, we run the experiment on an *Intel(R) Core i7-4500@1.80 GHZ 2.40 GHZ, 8 GB RAM 64-bit* and *Windows 8.1* OS. Table 7.17 shows that our high-level AR time (TpRO) averages 0.56 seconds. Thus, this timing makes our approach closer to real-time than existing solutions. Reported results in similar settings were found in the literature [49] with an average TpRO of 2.5 sec.

	Making cereal	Taking medicine	Stacking obj.	Unstacking obj.	Microwaving	Picking obj.(Bending)	Cleaning obj.	Takeout food	Arranging obj.	Eating meal	Null
Making cereal	1										
Taking medicine		1									
Stacking objects	.08		.59	.33							
Unstacking objects				1							
Microwaving					.59			.33			.08
Picking objects			.25			.67				.08	
Cleaning objects							1				
Takeout food							.08	.83		.08	
Arranging objects			.25	.08		.08			.59		
Eating meal										.92	.08

Table 7.13: Confusion matrix for high-level activities taking as input the sub-activities detected in the first stage tracking system

	Making cereal	Taking medicine	Stacking obj.	Unstacking obj.	Microwaving	Picking obj.(Bending)	Cleaning obj.	Takeout food	Arranging obj.	Eating meal	Null
Making cereal	1										
Taking medicine		1									
Stacking objects	.08		.59	.33							
Unstacking objects				.92					.08		
Microwaving					.83			.17			
Picking objects						1					
Cleaning objects							1				
Takeout food								1			
Arranging objects			.08			.25			.67		
Eating Meal										1	

Table 7.14: Confusion matrix for high-level activities taking as input the sub-activities 100% perfectly labelled from the CAD-120 dataset (ideal condition)

	Making cereal	Taking medicine	Stacking obj.	Unstacking obj.	Microwaving	Picking obj.(Bending)	Cleaning obj.	Takeout food	Arranging obj.	Eating meal	Null
Making cereal	.88			.13							
Taking medicine		.88									.13
Stacking objects			.17	.75							.08
Unstacking objects				1							
Microwaving	.83				.58			.08			.25
Picking objects					.08	.83		.83			
Cleaning objects							1				
Takeout food		.08			.58		.17	.17			
Arranging objects					.17	.5			.08		.25
Eating Meal					.08					0.67	.25

Table 7.15: Confusion matrix for high-level activities using a naïve approach: without applying heuristic filters for CAD-120 dataset.

Method	**Sub-Activities input**	**Precision**	**Recall**	**Accuracy**
Koppula et al. [128]	Full model with tracking	78.6 ± 4.1	78.3 ± 4.9	79.0 ± 4.7
Ours	Naïve approach without heuristic filters	64.1 ± 7.8	75.98 ± 13.53	61.36 ± 6.93
Ours	Real situation (predicted from 1st module-detection system)	84.1 ± 2.3	97.4 ± 0.66	82.9 ± 0.92
Ours	Ideal situation (labelled, 100% acc.)	90.8 ± 1.31	98.1 ± 1.25	91.07 ± 1.28

Table 7.16: Comparison of our approach with the dataset's algorithm

Activity Recognition Time	**Average Time**	**Standard Deviation Time**
Making cereal	1025.94	281.95
Taking medicine	212.9	38.01
Stacking objects	960.12	337.02
Unstacking objects	984.93	283.63
Microwaving food	400.15	229.32
Picking object (Bending)	234.13	301.46
Cleaning objects	480.5	197.79
Taking out food	333.84	249.54
Arranging objects	236.48	253.67
Having meal	733.78	224.85
Average	560.28	239.724

Table 7.17: Average recognition times (in milliseconds) per high-level activity.

Primitive Classes	*(define-primitive-concept User *top*)*
	*(define-primitive-concept Object *top*)*
	*(define-primitive-concept Activity *top*)*
	*(define-primitive-concept SubActivity *top*)*
Sub-activities	*(define-primitive-concept reaching SubActivity)*
	(define-primitive-concept moving SubActivity)
	(define-primitive-concept pouring SubActivity)
	(define-primitive-concept eating SubActivity)
	(define-primitive-concept drinking SubActivity)
	(define-primitive-concept opening SubActivity)
	(define-primitive-concept placing SubActivity)
	(define-primitive-concept closing SubActivity)
	(define-primitive-concept cleaning SubActivity)
	(define-primitive-concept null SubActivity)
High-level activities	*(define-primitive-concept cereal Activity)*
	(define-primitive-concept medicine Activity)
	(define-primitive-concept stacking Activity)
	(define-primitive-concept unstacking Activity)
	(define-primitive-concept microwaving Activity)
	(define-primitive-concept bending Activity)
	(define-primitive-concept cleaningObjects Activity)
	(define-primitive-concept takeout Activity)
	(define-primitive-concept arrangingObjects Activity)
	(define-primitive-concept eatingMeal Activity)
	(define-primitive-concept nullA Activity)
Object categories	*(define-primitive-concept kitchenware Object)*
	(define-primitive-concept stackable Object)
	(define-primitive-concept edible Object)
	(define-primitive-concept movable Object)
	(define-primitive-concept drinkingKitchenware Object)
	(define-primitive-concept pickable Object)
	(define-primitive-concept containerKitchenware Object)
	(define-primitive-concept arrangeable Object)

Table 7.18: Fuzzy concept definitions

Rule 1	*(define-concept antecedent1 (w-sum (0.17 reachMilkOrBowlOrBox)(0.41 moveMilkOrBowlOrBox)(0.24 placeMilkOrBowlOrBox)(0.01 openMilkOrBox)(0.16 pourMilkOrBox)))* *(define-concept consequent1 (g-and User (some performsActivity cereal)))*
Rule 2	*(define-concept antecedent2 (w-sum (0.29 reachCupOrMedicineBox)(0.3 moveCupOrMedicineBox)(0.1 placeCupOrMedicineBox)(0.1 openMedicineBox)(0.1 eatMedicineBox)(0.1 drinkCup)))* *(define-concept consequent2 (g-and User (some performsActivity medicine)))*
Rule 3	*(define-concept antecedent3 (w-sum (0.26 reachStackable)(0.27 moveStackable)(0.27 placeStackable)(0.20 nullSA)))* *(define-concept consequent3 (g-and User (some performsActivity stacking)))*
Rule 4	*(define-concept antecedent4 (w-sum (0.26 reachStackable)(0.27 moveStackable)(0.27 placeStackable)(0.20 nullSA)))* *(define-concept consequent4 (g-and User (some performsActivity unstacking)))*
Rule 5	*(define-concept antecedent5 (w-sum (0.32 reachMicroOrDrinkingKitchenware)(0.11 moveDrinkingKitchenware)(0.11 placeDrinkingKitchenware)(0.12 openMicro)(0.11 closeMicro)(0.23 nullSA)))* *(define-concept consequent5 (g-and User (some performsActivity microwaving)))*
Rule 6	*(define-concept antecedent6 (w-sum (0.26 reachPickable)(0.27 movePickable)(0.47 nullSA)))* *(define-concept consequent6 (g-and User (some performsActivity bending)))*
Rule 7	*(define-concept antecedent7 (w-sum (0.27 reachMicroOrCloth)(0.23 moveCloth)(0.1 placeCloth)(0.1 openMicro)(0.1 closeMicro)(0.1 cleanMicroOrCloth)(0.1 nullSA)))* *(define-concept consequent7 (g-and User (some performsActivity cleaningObjects)))*
Rule 8	*(define-concept antecedent8 (w-sum (0.38 reachContainerKitchenwareOrMicro)(0.12 moveContainerKitchenware)(0.12 placeContainerKitchenware)(0.13 openMicro)(0.13 closeMicro)(0.12 nullSA)))* *(define-concept consequent8 (g-and User (some performsActivity takeout)))*

Table 7.19: Fuzzy rules for each activity (I).

Rule 9	*(define-concept antecedent9 (w-sum (0.23 reachArrangeable) (0.27 moveArrangeable)(0.25 placeArrangeable)(0.25 nullSA)))* *(define-concept consequent9 (g-and User (some performsActivity arrangingObjects)))*
Rule 10	*(define-concept antecedent10 (w-sum (0.08 reachCup)(0.45 moveCup)(0.07 placeCup)(0.13 eatCup)(0.1 drinkCup)(0.17 nullSA)))* *(define-concept consequent10 (g-and User (some performsActivity eatingMeal)))*

Table 7.20: Fuzzy rules for each activity (II).

From the experiments conducted, it can be seen that activities such as *bending, stacking, microwaving* and *unstacking* are not as well recognized as the others. The propagation of errors from the first sub-activity recognition phase is a reason for this, as well as the noise in the detection of the objects and their positions (specially with respect to each other, and when occlusions appear). It is worth noticing that the AR is performed taking as input the output of the sub-activity recognition module that, at the same time, performs real-time (user and object) tracking. This adds an extra challenging dimension on the hybrid system, and therefore, the results presented are still promising. With the upcoming advances of a new generation of RGB-D sensors, these problems are expected to be solved, and therefore, heuristics used in the high-level AR phase are expected to produce better detection results.

Sub-activities definitions	*(define-concept openMilkOrBox (g-and User (some performsSubActivity (g-and opening (some usesObject (or milk box))))))* *(define-concept reachMilkOrBowlOrBox (g-and User (some performsSubActivity (g-and reaching (some usesObject (or milk bowl box))))))* *(define-concept moveMilkOrBowlOrBox (g-and User (some performsSubActivity (g-and moving (some usesObject (or box milk bowl))))))* *(define-concept placeMilkOrBowlOrBox (g-and User (some performsSubActivity (g-and placing (some usesObject (or box milk bowl))))))* *(define-concept pourMilkOrBox (g-and User (some performsSubActivity (g-and pouring (some usesObject (or milk box))))))* *(define-concept reachCup (g-and User (some performsSubActivity (g-and reaching (some usesObject cup)))))* *(define-concept reachMedicineBox (g-and User (some performsSubActivity (g-and reaching (some usesObject medicineBox)))))* *(define-concept openMedicineBox (g-and User (some performsSubActivity (g-and opening (some usesObject medicineBox)))))* *(define-concept moveMedicineBox (g-and User (some performsSubActivity (g-and moving (some usesObject medicineBox)))))* *(define-concept moveCup (g-and User (some performsSubActivity (g-and moving (some usesObject cup)))))* *(define-concept eatMedicineBox (g-and User (some performsSubActivity (g-and eating (some usesObject medicineBox)))))* *(define-concept placeCupOrMedicineBox (g-and User (some performsSubActivity (g-and placing (some usesObject (or cup medicineBox))))))* *(define-concept drinkCup (g-and User (some performsSubActivity (g-and drinking (some usesObject cup)))))* *(define-concept reachCupOrMedicineBox (g-and User (some performsSubActivity (g-and reaching (some usesObject (or cup medicineBox))))))* *(define-concept moveCupOrMedicineBox (g-and User (some performsSubActivity (g-and moving (some usesObject (or cup medicineBox))))))*

Table 7.21: Excerpt of fuzzy concepts used in the rules.

Sub-activities definitions	*(define-concept reachStackable (g-and User (some performsSubActivity (g-and reaching (some usesObject stackable)))))* *(define-concept moveStackable (g-and User (some performsSubActivity (g-and moving (some usesObject stackable)))))* *(define-concept placeStackable (g-and User (some performsSubActivity (g-and placing (some usesObject stackable)))))* *(define-concept reachMicro (g-and User (some performsSubActivity (g-and reaching (some usesObject microwave)))))* *(define-concept placeCloth (g-and User (some performsSubActivity (g-and placing (some usesObject cloth)))))* *(define-concept reachDrinkingKitchenware (g-and User (some performsSubActivity (g-and reaching (some usesObject drinkingKitchenware))))* *(define-concept moveDrinkingKitchenware (g-and User (some performsSubActivity (g-and moving (some usesObject drinkingKitchenware)))))* *(define-concept placeDrinkingKitchenware (g-and User (some performsSubActivity (g-and placing (some usesObject drinkingKitchenware)))))* *(define-concept openMicro (g-and User (some performsSubActivity (g-and opening (some usesObject microwave)))))* *(define-concept closeMicro (g-and User (some performsSubActivity (g-and closing (some usesObject microwave)))))* *(define-concept reachMicroOrDrinkingKitchenware (g-and User (some performsSubActivity (g-and reaching (some usesObject (or microwave drinkingKitchenware))))))* *(define-concept reachPickable (g-and User (some performsSubActivity (g-and reaching (some usesObject pickable)))))* *(define-concept movePickable (g-and User (some performsSubActivity (g-and moving (some usesObject pickable)))))* *(define-concept reachMicroOrCloth (g-and User (some performsSubActivity (g-and reaching (some usesObject (or microwave cloth))))))* *(define-concept moveCloth (g-and User (some performsSubActivity (g-and moving (some usesObject cloth)))))* *(define-concept cleanMicroOrCloth (g-and User (some performsSubActivity (g-and closing (some usesObject (or microwave cloth))))))*

Table 7.22: Excerpt of fuzzy concepts used in the rules (Part II).

Sub-activities definitions	*(define-concept cleanMicro (g-and User (some performsSubActivity (g-and cleaning (some usesObject microwave)))))* *(define-concept cleanCloth (g-and User (some performsSubActivity (g-and cleaning (some usesObject cloth)))))* *(define-concept reachContainerKitchenwareOrMicro (g-and User (some performsSubActivity (g-and reaching (some usesObject (or microwave containerKitchenware))))))* *(define-concept moveContainerKitchenware (g-and User (some performsSubActivity (g-and moving (some usesObject containerKitchenware)))))* *(define-concept placeContainerKitchenware (g-and User (some performsSubActivity (g-and placing (some usesObject containerKitchenware)))))*

Table 7.23: Excerpt of fuzzy concepts used in the rules (Part III).

Chapter 8

Contributions

> Solo el misterio nos hace vivir,
> solo el misterio
>
> Federico García Lorca

This PhD thesis provides a set of contributions that can be summarized while considering different points of view. On the more theoretical, modelling side, two main ontologies were proposed, and work was done within an extra security and privacy ontology:

- An OWL 2 3D depth sensors crisp ontology ($\mathcal{ALC}$ DL expressivity) composed of 164 classes, 53 object properties, 58 data properties and 93 individuals, and based on the Kinect for Windows API. The structure of the ontology is based on Kinect Natural User Interface, Kinect Interaction, Fusion and Audio modules. We exemplified the usage of the proposed ontology with different domain examples, such as remote rehabilitation, physical exercises and long-term changes in posture.

- A Fuzzy OWL 2 ontology for generic activity recognition and a use case on the work/office domain, validated with HermiT 1.3.6, Pellet 2.3.0 (2.2.0 Protégé plug-in) OWL reasoners and *fuzzyDL 1.1.* It consists of 228 classes, 133 object properties, 62 data properties and 33 test individuals, with $\mathcal{SROIQ}(\mathbf{D})$ DL expressiveness. Example queries and rules were more extensively tested with a small and a larger complex human activity datasets.

- Automatic treatment of uncertain, incomplete and vague information was tackled with a fuzzy ontology that makes the system (sensor and prediction) fault tolerant for human activity modelling and recognition.

On the development and implementation real-case side, we contributed a hybrid system that integrates the favourable features typical of machine learning with the ability to include semantics and improve the model interpretability, typical of using knowledge engineering methods. The tracking and recognition system of complex sub-activities and high-level activities considers object interaction from 3D depth images and is validated with a public, novel, real life and non-synthetic dataset of challenging complex activities. The dataset is labelled uniquely with 3D depth data (i.e., not requiring to carry wearable sensors). It is composed of a continuous video data stream (requiring tracking), plus a discrete data stream of sub-activities for the second phase. The results are the following:

- Tracking and recognition of complex sub-activities and high-level activities involving object interaction from 3D-depth images in a novel, real life and non-synthetic dataset. Although there exist many crisp ontologies for activity modelling such as mIO! [165], PalSPOT [178], CONON [222], PiVOn [99] or Situation Ontology [225], there does not exist a fuzzy ontology previous to ours [61], dedicated to activity recognition. Therefore, apart from validating the advantages of our fuzzy ontology, at a theoretical level, previously in [61], in this work, we took a step further. We developed a hybrid system applied to practical level and used a more complex real life dataset.

- Statistically significant improved results on precision, recall and accuracy of 91.5%, 97% and 90.1%, respectively, for the first data-driven stage of the AR system (sub-activities). We achieve 84.1% in precision, 97.4% in recall and 82.9% in accuracy for the knowledge-driven, ontological and last stage of the AR system, i.e., for high-level activities. Assuming an ideal scenario with 100% correctly labelled input sub-activities, a precision of 90.8%, a recall of 98.1% and an accuracy of 91.07% are achieved.

- Automatic treatment of uncertain, incomplete and vague information with a fuzzy ontology that makes the system (sensor and prediction) fault tolerant. While a crisp system considers axioms true or false in its totality, in a fuzzy system, conditions and facts can occur to a certain degree in [0,1]. This means that we can have degrees of truth for each isolated event, sensor reading, interpretation, and build on top of that, considering the uncertainty of each data source independently, but in relation to the satisfaction of high-level concepts. This is handled in our hybrid system thanks to the higher abstraction inference layer through *fuzzyDL* reasoner. For instance, when any of the sensors used for data acquisition fails or breaks, the activity will not be recognized at all in purely crisp approaches. In contrast to traditionally crisp approaches, uncertainty reasoning provides with a lower degree of recognition in [0,1] that still gives a degree of truth for the activity performed, even if a sensor fails, breaks, or its reading is not captured. Faults can also occur in data interpretation (e.g. recognizing objects or users), and thus, this is another kind of tolerated *fault*.

- A method that simplifies complexity in the training phase. In case of new addition/removal/replacement of input data, instead of re-training, it is only required to modify the affected activity rule, since the activities and interrelations can be modelled as common-sense rules. It provides interpretability and readability of rules versus black-box machine learning approaches by facilitating the conceptualization of common-sense knowledge, context-awareness and object interaction.

- Reactiveness and scalability: Real-time tracking for on-line recognition and deployment in, e.g., AmI or AAL environments. For instance, monitoring and recognition of activities (e.g. to assist independent elders) can be done in an average time of 0.56 seconds in the executed experiment.

The integration of our fuzzy ontology proposal into a hybrid data and knowledge driven architecture thus, resulted on a versatile system with varied (from fine to coarse-grained) levels of abstraction to detect atomic and complex activities, considering the user's interaction with objects, and real-time tracking as well as uncertain or imprecise data, such as missing sensor observations or execution variations. These features are tackled for first time taking into account, ontologically, the semantics of activities, sub-activities and real-time object interaction [60].

On the architectural, deployment and user experience side, the contributions are the following:

- An architecture model with a mapping between crisp and fuzzy OWL to allow query federation among crisp and fuzzy systems, allowing to take advantage of both architectural and semantic paradigms. The proposed architecture (in contribution paper 7 [66]), the crisp to fuzzy-language extension and the support for fuzzy reasoning show the path for dealing with current issues on SSs' usability, as well as for setting the base for precise, and at the same time flexible, personalized and adaptive Smart Spaces.

- A prototype user interface for general purpose visual programming of Smart Spaces where an underlying semantic model is kept to allow ontological reasoning and interoperable programming. The visual language model mock-up proposal, can as well serve as an educative interface for teaching basic SW technologies and logic programming ideas intuitively. However, as a general purpose visual language based on a semantic metamodel, we support rule composition for rapid development of mash-up applications under the presence of imprecision or uncertainty. This contribution follows visual language design guidelines [89] for an intuitive, *well matched*, visual language, i.e., its representation clearly captures the key features of the represented artefact (in our case RDF triples), in addition to simplify various desired reasoning tasks (i.e., hiding namespaces, query languages, fuzzy logic and Semantic Web formalisms).

Chapter 9

Further remarks and future work

> All things are so very uncertain,
> and that's exactly what makes
> me feel reassured.
>
> Tove Jansson

Current trends show that tracking and monitoring people is becoming an integral part of everyday life. Data-driven approaches (HMM, Bayesian networks, decision trees, etc.) appear to stand out in the literature, in contrast to the newly emergent knowledge-based techniques. The latter include, among others, information indexing and retrieval, hierarchical knowledge sources -taxonomies, ontologies, encyclopaedia, dictionaries-, representation languages, distributed knowledge, and logical or KB tools. This thesis focused on knowledge-based techniques, more concretely, ontologies.

One of the research questions was to find whether we can effectively use semantic and ontology-based reasoning to recognize different level, simple and complex, real life human activities. With this aim, we contributed with a survey (in Chapter 3) on available techniques for human behaviour recognition. We proposed an evaluation taxonomy for learning procedures, methods, models, and modelling capabilities on data-driven and knowledge-based approaches.

In contrast to data-driven classical methods, we conclude that knowledge-based techniques, such as ontology-based activity modelling, add a set of advantages for incremental and context-aware recognition. Ontologies are suitable approaches to achieve interoperability, abstraction, and modularity in an easy way. Another advantage of ontologies, and in general, knowledge-based methods, is that expert knowledge can be introduced directly in the knowledge base, while data-driven approaches require a great amount of initial data, training of the model, and validation. This fact, therefore, makes necessary the inclusion of common sense knowledge in the ontology in order to palliate the need for large amounts of data. This can be seen as an advantage when there is not enough training data.

As we decided to further explore and exploit the ontological branch of AR, we reviewed the state-of-the art on ontologies for activity recognition and analysed them from different perspectives. We identified some missing capabilities and sub-domains. The main problem with existing ontologies is that they are unable to model and handle automatically uncertainty, vague or incomplete knowledge. Therefore, we used the spotted missing features to create a novel fuzzy ontology to deal with uncertainty. In this sense, we facilitated the way of representing more flexible activity models. Incomplete data such as missing sensor readings due to occlusions or sensor failures, was treated in this way naturally.

In order to answer the question *Is the semantic-fuzzy framework integrable with a traditional data-driven system for activity recognition, and able to improve the context-awareness interpretability, looseness and accuracy/-precision of traditional methods?*, we proposed a hybrid semantic recognition module for real-time tracking and recognition of activities of different granularity, accounting for imprecision, context-awareness and object interaction.

This data- and knowledge-driven 2-phased algorithm handles the challenges of complex AR systems by using the most suitable method in each phase, real-time speed and accuracy for the first stage, and uncertainty treatment, as well as the provision of contextual meaning for activities, in the second phase. The system entails an improvement over both, entirely data-driven approaches and merely ontology-based approaches. It was so demonstrated that the importance of embedding machine learning techniques into emerging research on knowledge-driven approaches is crucial towards activity modelling and recognition in complex scenarios. The hybrid approach, together with the developed ontologies, also showed how an appropriate combination of computer vision algorithms with semantic models of human movement and interaction can significantly improve context-awareness, recognition accuracy and activity analysis precision.

The purpose of the filters applied in the knowledge-based phase are indispensable in achieving good accuracy and precision results, since their main aim is helping to discriminate among similar activities that use similar objects, or activities that use almost the same subsequences of sub-activities in time. These filters form part of the hybrid activity recognition method based on knowledge engineering. This manual specification is a typical characterization required a priori in knowledge engineering methods in general, where experts are required to set domain rules for an expert system to produce successful predictions.

Therefore, we can identify some disadvantages in our method. It is not a fully automatic method, since it requires the inclusion of expert knowledge in the KB. However, in some applications this may not be a limitation, since it is possible to define ad-hoc activities, and in this way, reduce the necessary training time and the adaptation of the system to different environments. In terms of software design, an inconvenience of the proposal is its complexity compared to methods that only use either knowledge-based or data-driven approaches, for learning and recognizing activities. However, this complexity increase is compensated with an improvement in the activity recognition rate, as we have seen in the experimental section. Finally, the use of a fuzzy ontology to model knowledge, as opposed to traditional crisp ontologies, can better model the problem and the uncertainty in activities; however, the higher order of complexity of fuzzy reasoners can be a limitation compared to crisp reasoners, when the number of activities in the ontology grow substantially within the Knowledge Base, as it can require a processing time much greater for the recognition. In [61] we analysed theoretically this problem for cases up to KB sizes of 10^5 triples, where crisp reasoning was compared with fuzzy reasoning, and found that for the current application would be feasible to use these techniques, obtaining reduced run times, whenever the system is applied on a limited space such as a home setting. In fact, despite this limitation, in this work we found that in these cases it is possible for the system run time to become close to real time.

If the system needs to recognize new activities, a new class for each sub-activity and activity to be recognized need to be added to the ontology definition. This is the alternative that knowledge-based methods offer against data-driven ones. It is understood that the former are not totally ”training-free” methodologies and that they come with a compromise, which we believe it is small, once the data-driven training phase has been achieved within a hybrid system. In order to upgrade knowledge-driven AR systems such as ours, the updating of sub-activity weights and rule definitions in the ontology is necessary. We believe this is an affordable trade-off in contrast to recording new datasets with diverse users, objects, old and new activities, plus retraining and validating the system.

By integrating the fuzzy ontology into a more complex hybrid data and knowledge-driven architecture, it was tested with a real life dataset with complex activities. As a result, we provided the possibility of representing imprecise and uncertain information for more accurate modelling of everyday human tasks and human natural language. As a consequence, the gap between end-users and the intelligent systems that govern the Smart Space can be bridged.

Regarding the deployment phase, we approached the challenges required to allow the development of context-aware intelligent applications for Smart Space-based infrastructure. The goal was to model and process context information using our development tool and Nokia's Smart-M3 architecture.

We proposed an adaptable and scalable ontology-based ubiquitous computing framework and rule-based reasoning to infer high-level context. The approach deals with key issues in context-aware ubiquitous computing such as being adaptive and proactive to changes in the environment, incorporation of novel sources of context information and automatic code generation from the context ontology to provide seamless interoperability. When the target is an end-user with no technical knowledge, we also presented a general purpose programming visual language model that pushes the devised evolution of the SW from a data modelling to a computational medium [181], by bringing the advantages of the SW closer to any non expert user. This thesis' applications thus, range widely from AAL and eHealth to home automation, industry processes or Smart Hospitals. Regarding the latter area, our work done on nursing and medication tracking processes at hospital wards [120, 65, 95] could as well benefit from this thesis' HAR proposal, as for next steps on automated monitorization.

However, activity recognition challenges on multi-user settings still remain to be tackled as continuation to the work presented. Equally important is to further generalize the system to recognize concurrent/interleaved activities, which were not included in the CAD-120 dataset.

A future challenge to deal with is achieving adjustable processes of ontology evolution for adapting to natural changes in human behaviour and/or environments. Adaptive activity modelling, such as the semi-automatic model in [50], evolves from initial *seed* activity models through a continuous activity discovery, by using pattern recognition and ontology-based learning and reasoning. Another future direction to explore is modelling and detecting human behaviour changes. Using *learning instance matching*, i.e., (data level, non-schema) ontology mapping for new data integration [216], could be an approach towards automating the evolution or learning of behaviour changes. [23] propose a novel clustering process, in this direction, to make ontological models evolve in time for user behaviour adaptation. By means of action clusters, varying ways of performing activities are learnt to acquire specialized activity models.

In relation with change discovery, it would be interesting to consider: a) changes in a frequent behaviour or environment migration (anomaly detection [115, 19, 233, 187]) and b) when a predefined model of behaviour is performed by a different user than the one it was designed for. Works such as [157], focused on ontology evolution process, could be a starting point. The change in an ontology consists here of six phases: change capturing, change representation, semantics of change, change implementation, change propagation, and change validation. This strategy allows activity learning and model evolution through ontology-based activity traces. It also makes evident that activity models are key to support reusable, adaptive, and personalized activity recognition, as well as to improve scalability and applicability of any assistive system.

Emergent reasoners for support with uncertainty reasoning shall be evaluated to accommodate all requirements [226]. Future work should also consider the use of equally expressive fuzzy reasoners such as *fuzzyDL*, with the added value of supporting retractability of axioms [209]. This means the ability to allow the deletion of data from the KB in order to update information, since AAL makes use of rapidly changing data sources. This addition would permit the reduction of the computational complexity required in the current approach, in order to empty the database every time a query needs to be done in a different time window. As in Description Logics, a given axiom's certainty value can only be raised and not lowered, with the addition of new axioms, due to the monotonicity property of this type of first order logic, it is unavoidable to empty and reload the KB every time a new time window is considered. This is required to preserve more coherent results and favour the last events occurred (despite the sliding window's flexible size). A compromise between expressibility, efficiency for large numbers of events, and precision must be seek within reasoners. Only in this way will the power, but also the functionality of automatic uncertainty reasoning be preserved and fully taken advantage of.

Concerning infrastructural issues, in the future, scalability and performance of the proposed hybrid crisp-fuzzy architecture should be studied, as well as possible alternatives against data redundancy (due to having dual crisp and fuzzy KBs), apart from managing consistency (e.g., double update synchronization) in the joint KB. The proposed architecture, with the crisp-to-fuzzy language extension and the support for fuzzy reasoning, show the path for dealing with current issues on SSs' usability as well as for setting the base for precise, and at the same time flexible, personalized and adaptive Smart Spaces.

With reference to the end-user programmability of the SS, future work should be focused on the implementation of the mockup GUIs, accounting for two aspects; first, having as focus the reactive rule editing, for customizing the behaviour of the SS, and second, deploying a tool for designing/specifying human behaviours by the end-user (e.g, for a caregiver to monitor an elder remotely). This tool should easily provide, following our graphical model, a rapid mechanism to define, for instance, the behaviour *taking the dog for a walk* or *take grandma to the bank*.

Other scenarios than the considered ones in AAL should be explored as well, in more concrete cases than ADLs for, e.g., the growing population or the elderly. For instance, more concrete domain target groups can be the focus for more specific assistance, e.g., people with autism or epilepsy. These are just a couple of examples, but any other (e.g., industrial) process where automatic monitoring could help following or tracking strict guidelines, could benefit from the use of activity modelling and recognition, and could further improve or refine the methods and framework proposed in this thesis. Ultimately, the technology transfer to society remains as left work to close the cycle and help in different scenarios, from elderly remote monitoring to tele-rehabilitation.

Bibliography

[HAR] Fuzzy Human Behaviour Ontology, rules and queries: [online]: http://users.abo.fi/ndiaz/public/fuzzyhumanbehaviourontology/.

[Fuz] FuzzyOWL Plugin. [online]: http://gaia.isti.cnr.it/ straccia/software/fuzzyowl/fuzzyowl2.1.1.zip.

[Kin] Kinect Ontology [online]: http://users.abo.fi/rowikstr/kinectontology/.

[OWL] OWL-S Services: http://www.daml.org/services/owl-s/.

[owl] OWLIM-SE. Semantic repository for RDF(S) and OWL. OWLIM-SE (standard edition) v4.3. [online]: http://owlim.ontotext.com/display/owlimv43/owlim-se.

[PyL] Pythologic, Prolog Syntax in Python: http://code.activestate.com/recipes/303057-pythologic-prolog-syntax-in-python/.

[Ont] Smart-M3 Ontology to Python API Generator: http://sourceforge.net/projects/smart-m3/files/smart-m3-ontology_to_python-api_generator_v0.9.1beta.tar.gz/.

[spi] SPIN [online]: http://www.spinrdf.org/.

[swr] SWRL: A Semantic Web Rule Language Combining OWL and RuleML [online]: http://www.w3.org/submission/swrl/.

[10] (Feb. 2010). W3C Group: Task Meta Models. http://www.w3.org/2005/Incubator/model-based-ui/wiki/Task_Meta_Models.

[11] Aarts, E. and Eggen, B. (2002). Ambient intelligence in homelab.

[12] Abdualrazak, B., Malik, Y., and Yang, H.-I. (2010). A taxonomy driven approach towards evaluating pervasive computing system. In *Proceedings of the Aging friendly technology for health and independence, and 8th international conference on Smart homes and health telematics*, ICOST'10, pages 32–42, Berlin, Heidelberg. Springer-Verlag.

[13] Acampora, G. and Loia, V. (2005). Fuzzy control interoperability and scalability for adaptive domotic framework. *IEEE Transactions on Industrial Informatics*, 1:97 – 111.

[14] Aggarwal, J. and Ryoo, M. (2011). Human activity analysis: A review. *ACM Computing Surveys.*, 43(3):16:1–16:43.

[15] Agostini, A., Bettini, C., and Riboni, D. (2009). Hybrid reasoning in the CARE middleware for context awareness. *Int. J. Web Eng. Technol.*, 5(1):3–23.

[16] Alessandro Ciaramella, Mario G.C.A. Cimino, F. M. U. S. (2010). Combining fuzzy logic and semantic web to enable situation-awareness in service recommendation. In *Proceedings of the 21st International Conference on Database and Expert Systems Applications (DEXA-10)*, volume 6261 of *Lecture Notes in Computer Science*, pages 31–45. Springer Verlag.

[17] Amailef, K. and Lu, J. (2013). Ontology-supported case-based reasoning approach for intelligent m-government emergency response services. *Decision Support Systems*, 55(1):79 – 97.

[18] Amoretti, M., Wientapper, F., Furfari, F., Lenzi, S., and Chessa, S. (2010). Sensor data fusion for activity monitoring in ambient assisted living environments. In Hailes, S., Sicari, S., Roussos, G., Akan, O., Bellavista, P., Cao, J., Dressler, F., Ferrari, D., Gerla, M., Kobayashi, H., Palazzo, S., Sahni, S., Shen, X. S., Stan, M., Xiaohua, J., Zomaya, A., and Coulson, G., editors, *Sensor Systems and Software*, volume 24 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 206–221. Springer Berlin Heidelberg.

[19] Anderson, D. T., Ros, M., Keller, J. M., Cuéllar, M. P., Popescu, M., Delgado, M., and Vila, A. (2012). Similarity measure for anomaly detection and comparing human behaviors. *International Journal of Intelligent Systems.*, 27(8):733–756.

[20] Arce, F., Mario, J., and Valdez, G. (2010). Accelerometer-based hand gesture recognition using artificial neural networks. *Soft Computing*, 318:67–77.

[21] Atallah, L. and Yang, G.-Z. (2009). The use of pervasive sensing for behaviour profiling, a survey. *Pervasive and Mobile Computing*, 5(5):447 – 464.

[22] Augusto, J. (2007). Ambient intelligence: The confluence of pervasive computing and artificial intelligence. In Schuster, A., editor, *Intelligent Computing Everywhere*, pages 213–234. Springer Berlin Heidelberg.

[23] Azkune, G., Almeida, A., de Ipiña, D. L., and Chen, L. (2015). Extending knowledge-driven activity models through data-driven learning techniques. *Expert Systems with Applications*, 42(6):3115 – 3128.

[24] Baader, F. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press.

[25] Baader, F., Horrocks, I., and Sattler, U. (2007). Description logics. In van Harmelen, F., Lifschitz, V., and Porter, B., editors, *Handbook of Knowledge Representation*, pages 135–179. Elsevier.

[26] Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). A Survey on Context-Aware systems. In *International Journal of Ad Hoc and Ubiquitous Computing*, volume 2, Geneva, Switzerland. Inderscience Publishers.

[27] Baumgartner, N. and Retschitzegger, W. (2006). A survey of upper ontologies for situation awareness. *Proceedings of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering, St. Thomas, US VI*, pages 1–9+.

[28] Berg, A., Karvinen, P., Grönroos, S., Wickström, F., Díaz Rodríguez, N., Hosseinzadeh, S., and Lilius, J. (2013). A Scalable Distributed M3 Platform on a Low-Power Cluster. In Soininen, J.-P., Balandin, S., Lilius, J., Liuha, P., and Cinotti, T. S., editors, *Open International M3 Semantic Interoperability Workshop* , volume 21 of *TUCS Proceedings*, page 49– 58. TUCS.

[29] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American.*

[30] Bettini, C., Pareschi, L., and Riboni, D. (2008). Efficient profile aggregation and policy evaluation in a middleware for adaptive mobile applications. *Pervasive and Mobile Computing*, 4(5):697–718.

[31] Bobillo, F. (2008). *Managing Vagueness in Ontologies.* PhD thesis.

[32] Bobillo, F., Delgado, M., and Gómez-Romero, J. (2012). DeLorean: A reasoner for fuzzy OWL 2. *Expert Systems with Applications*, 39(1):258 – 272.

[33] Bobillo, F. and Straccia, U. (2008). *fuzzyDL*: An expressive fuzzy description logic reasoner. In *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages 923–930. IEEE Computer Society.

[34] Bobillo, F. and Straccia, U. (2011). Fuzzy ontology representation using OWL 2. *Int. J. Approx. Reasoning*, 52(7):1073–1094.

[35] Boger, J., Hoey, J., Poupart, P., Boutilier, C., Fernie, G., and Mihailidis, A. (2006). A planning system based on Markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology in Biomedicine*, 10:323 –333.

[36] Bohannon, R. (2006). Reference values for the five-repetition sit-to-stand test: a descriptive meta-analysis of data from elders. *Percept Mot Skills*, 1(103):215–22.

[37] Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* PhD thesis, Institute for Telematica and Information Technology, University of Twente, Enschede, The Netherlands.

[38] Brdiczka, O., Reignier, P., and Crowley, J. (2007). Detecting individual activities from video in a smart home. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, page 363–370.

[39] Brush, A., Scott, J., and Krumm, J. (2010). Activity recognition research: The good, the bad, and the future. In *Pervasive 2010 Workshop.*

[40] Cantera-Fonseca, J. and Lewis, R. (June 2009.). Delivery Context Ontology. W3C Working Draft.

[41] Carlsson, C., Brunelli, M., and Mezei, J. (2010). Fuzzy ontologies and knowledge mobilisation: Turning amateurs into wine connoisseurs. In *FUZZ-IEEE*, pages 1–7. IEEE.

[42] Casas, R., Blasco Marín, R., Robinet, A., Delgado, A. R., Yarza, A. R., Mcginn, J., Picking, R., and Grout, V. (2008). User modelling in ambient intelligence for elderly and disabled people. In *Proceedings of the 11th international conference on Computers Helping People with Special Needs*, ICCHP '08, pages 114–122, Berlin, Heidelberg. Springer-Verlag.

[43] Chaaraoui, A. A., Climent-Pérez, P., and Flórez-Revuelta, F. (2012). A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, 39(12):10873 – 10888.

[44] Chen, H., Finin, T., and Joshi, A. (2003a). An Ontology for context-aware pervasive computing environments. In *Proceedings of the Workshop on Ontologies in Agent Systems.*

[45] Chen, H., Finin, T., and Joshi, A. (2003b). Semantic web in a pervasive context-aware architecture. In *Artificial Intelligence in Mobile System (AIMS 2003), In conjunction with UBICOMP*, pages 33–40.

[46] Chen, H., Finin, T., and Joshi, A. (2005). The SOUPA ontology for pervasive computing. In Tamma, V., Cranefield, S., Finin, T., and Willmott, S., editors, *Ontologies for Agents: Theory and Experiences*, Whitestein Series in Software Agent Technologies, pages 233–258. Birkhäuser Basel.

[47] Chen, L., Nugent, C., Mulvenna, M., Finlay, D., Hong, X., and Poland, M. (2008). Using event calculus for behaviour reasoning and assistance in a smart home. In Helal, S., Mitra, S., Wong, J., Chang, C., and Mokhtari, M., editors, *Smart Homes and Health Telematics*, volume 5120 of *Lecture Notes in Computer Science*, pages 81–89. Springer Berlin Heidelberg.

[48] Chen, L. and Nugent, C. D. (2009). Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems (IJWIS)*, 5(4):410–430.

[49] Chen, L., Nugent, C. D., and Wang, H. (2012). A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974.

[50] Chen, L., Okeyo, G., Wang, H., Sterritt, R., and Nugent, C. D. (2011). A systematic approach to adaptive activity modeling and discovery in smart homes. In *4th International Conference on Biomedical Engineering and Informatics (BMEI)*, pages 2192–2196.

[51] Chen, L. L. and Biswas, J. (2009). Tutorial: an introduction to ontology-based activity recognition. Proceedings of the 7th International Conference on Mobile Computing and Multimedia (MoMM2009).

[52] Cheng, J., Ma, Z., and Yan, L. (2010). f-SPARQL: A flexible extension of SPARQL. In Bringas, P., Hameurlain, A., and Quirchmayr, G., editors, *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 487–494. Springer Berlin Heidelberg.

[53] Choi, W., Pantofaru, C., and Savarese, S. (2011). Detecting and tracking people using an RGB-D camera via multiple detector fusion. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1076 –1083.

[54] Cook, D. J., Augusto, J. C., and Jakkula, V. R. (2007). Ambient intelligence: Technologies, applications, and opportunities.

[55] Cook, D. J., Augusto, J. C., and Jakkula, V. R. (2009). Review: Ambient intelligence: Technologies, applications, and opportunities. *Pervasive Mobile Computing*, 5(4):277–298.

[56] Cook, D. J. and Das, S. K. (2005). *Smart Environments: Technology, Protocols and Applications.* Wiley series on parallel and distributed computing. John Wiley.

[57] Corradini, A. (2001). Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 82–89.

[58] Crandall, A. and Cook, D. J. (2010). *Learning Activity Models for Multiple Agents in a Smart Space*, page 751.

[59] d'Aquin, M. and Noy, N. F. (2012). Where to publish and find ontologies? A survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11(0):96 – 111.

[60] Díaz Rodríguez, N., Cadahía, O. L., Cuéllar, M. P., Lilius, J., and Calvo-Flores, M. D. (2014a). Handling real-world context awareness, uncertainty and vagueness in real-time human activity tracking and recognition with a fuzzy ontology-based hybrid method. *Sensors*, 14(10):18131–18171.

[61] Díaz Rodríguez, N., Cuéllar, M. P., Lilius, J., and Delgado Calvo-Flores, M. (2014b). A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66(0):46 – 60.

[62] Díaz Rodríguez, N., Cuéllar, M. P., Lilius, J., and Delgado Calvo-Flores, M. (2014c). Semantic and fuzzy modelling of human behaviour in smart spaces. a case study on ambient assisted living and remote rehabilitation. In *Bailón-Moreno, R. (dir.) I Congreso Nacional Multidisciplinar de Jóvenes Investigadores, I CNMJI (The National Multidisciplinary Congress of Young Researchers). Granada (Spain). Feb 19-21.*

[63] Díaz Rodríguez, N., Grönroos, S., Wickström, F., Karvinen, P., Berg, A., Hosseinzadeh, S., Karppi, M., and Lilius, J. (2013a). M3 Interoperability for Remote Rehabilitation with Kinect. In Soininen, J.-P., Balandin, S., Lilius, J., Liuha, P., and Cinotti, T. S., editors, *Open International M3 Semantic Interoperability Workshop*, volume 21, page 153–163. TUCS Lecture Notes.

[64] Díaz Rodríguez, N., Lilius, J., Cuéllar, M. P., and Delgado Calvo-Flores, M. (2013b). Extending Semantic Web Tools for Improving Smart Spaces Interoperability and Usability. In Omatu, S., Neves, J., Rodriguez, J. M. C., Paz Santana, J. F., and Gonzalez, S. R., editors, *Distributed*

Computing and Artificial Intelligence, volume 217 of *Advances in Intelligent Systems and Computing*, pages 45–52. Springer International Publishing.

[65] Díaz Rodríguez, N., Lilius, J., Danielsson-Ojala, R., Pirinen, H., Kauhanen, L., Salanterä, S., Majors, J., Björlund, S., Rautanen, K., Salakoski, T., and Tuominen, I. (2014d). Can IT health-care applications improve the medication tray-filling process at hospital wards? an exploratory study using eye-tracking and stress response. In *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom 2014)*, Natal, Brazil.

[66] Díaz Rodríguez, N., Lilius, J., P. Cuéllar, M., and Delgado Calvo-Flores, M. (2013c). An approach to improve semantics in Smart Spaces using reactive fuzzy rules. In Pedrycz, W. and Reformat, M. Z., editors, *IFSA World Congress - NAFIPS Annual Meeting (IFSA/NAFIPS)*, pages 436–441, Edmonton, Canada. IEEE.

[67] Díaz Rodríguez, N., Wikström, R., Grönroos, S., Wickström, F., Karvinen, P., Berg, A., Hosseinzadeh, S., Lilius, J., Cuéllar, M. P., Delgado Calvo-Flores, M., and Karppi, M. (2014e). Monitorizado remoto de ejercicios de rehabilitación en tiempo real con sensores de profundidad 3d y tecnologías semánticas. In *In Bailón-Moreno, R. (dir.) I Congreso Nacional Multidisciplinar de Jóvenes Investigadores, I CNMJI (The National Multidisciplinary Congress of Young Researchers). Granada (Spain). Feb 19-21.*

[68] Díaz Rodríguez, N., Wikström, R., Lilius, J., Cuéllar, M. P., and Delgado Calvo Flores, M. (2013d). Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors. In Urzaiz, G., Ochoa, S., Bravo, J., Chen, L., and Oliveira, J., editors, *Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction*, volume 8276 of *Lecture Notes in Computer Science*, pages 254–261. Springer International Publishing.

[69] Delgado, M., Ros, M., and Vila, A. (2009). Correct behavior identification system in a tagged world. *Expert Systems with Applications*, 36(6):9899 – 9906.

[70] Dey, A. K. and Abowd, G. D. (2000). The Context Toolkit: Aiding the development of context-aware applications. In *Workshop on Software Eng. for Wearable and Pervasive Computing.*

[71] Díaz Rodríguez, N., Cuéllar, M. P., Lilius, J., and Delgado Calvo-Flores, M. (2014). A survey on ontologies for human behavior recognition. *ACM Comput. Surv.*, 46(4):43:1–43:33.

[72] Díaz Rodríguez, N., Kankaanpää, P., Saleemi, M. M., Lilius, J., and Porres, I. (2012). Programming biomedical smart space applications with BioImageXD and PythonRules. In *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, SWAT4LS '11, pages 10–11, New York, NY, USA. ACM.

[73] Díaz Rodríguez, N., Lilius, J., Pegalajar Cuéllar, M., and Delgado Calvo-Flores, M. (2013). Rapid prototyping of semantic applications in smart spaces with a visual rule language. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, UbiComp '13 Adjunct, pages 1335–1338, New York, NY, USA. ACM.

[74] Doctor, F., Hagras, H., and Callaghan, V. (2005). A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(1):55–65.

[75] Dubois, D. and Prade, H. (2001). Possibility theory, probability theory and multiple-valued logics: A clarification. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):35–66.

[76] Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., and Burgelman, J.-C. (2001). Scenarios for ambient intelligence in 2010.

[77] Dutta, S., Chatterjee, A., and Munshi, S. (2009). An automated hierarchical gait pattern identification tool employing cross-correlation-based feature extraction and recurrent neural network based classification. *Expert Systems*, 26(2):202–217.

[78] Ejigu, D., Scuturici, M., and Brunie, L. (2007). An ontology-based approach to context modeling and reasoning in pervasive computing. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 14 –19.

[79] Ermes, M., Parkka, J., Mantyjarvi, J., and Korhonen, I. (2008). Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20 –26.

[80] Foukarakis, M. (2009). *Informational system for managing photos and spatial information using sensors, ontologies and semantic maps.* PhD thesis, Technical University of Crete.

[81] Gharsellaoui, A., Bellik, Y., and Jacquet, C. (2012). Requirements of Task Modeling in Ambient Intelligent Environments. In *AMBIENT 2012, the Second International Conference on Ambient Computing, Applications, Services and Technologies*, pages 71–78. IARIA.

[82] Gómez-Romero, J., Patricio, M. A., García, J., and Molina, J. M. (2011). Ontology-based context representation and reasoning for object tracking and scene interpretation in video. *Expert Systems with Applications*, 38(6):7494–7510.

[83] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220.

[84] Gu, T., Pung, H. K., and Zhang, D. Q. (2004). A middleware for building context-aware mobile services. In *Proceedings of IEEE Vehicular Technology Conference*.

[85] Gu, T., Wu, Z., Tao, X., Pung, H. K., and Lu, J. (2009). epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition. In *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, PerCom '09, pages 1–9, Washington, DC, USA. IEEE Computer Society.

[86] Guarino, N., Oberle, D., and Staab, S. (2009). What is an ontology? In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, volume XIX of *International Handbooks on Information Systems*. Springer.

[87] Guenterberg, E., Yang, A., Ghasemzadeh, H., Jafari, R., Bajcsy, R., and Sastry, S. (2009). A method for extracting temporal parameters based on hidden Markov models in body sensor networks with inertial sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 13(6):1019 –1030.

[88] Guo, C., Li, H., and Pan, D. (2010). An improved piecewise aggregate approximation based on statistical features for time series mining. In *Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management*, KSEM'10, pages 234–244, Berlin, Heidelberg. Springer-Verlag.

[89] Gurr, C. (2006). Computational diagrammatics: diagrams and structure. In Besnard, D., Gacek, C., and Jones, C. B., editors, *Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective*. Springer London.

[90] Gurr, C. (2007). Visualizing a logic of dependability arguments. In Cox, P., Fish, A., and Howse, J., editors, *Visual Languages and Logic Workshop*

(VLL 2007), volume 274, pages 97–109. Workshop within IEEE Symposium on Visual Languages and Human Centric Computing VL/HCC 07.

[91] Haarslev, V., Pai, H.-I., and Shiri, N. (2007). Optimizing tableau reasoning in ALC extended with uncertainty. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June.*

[92] Habiballa, H. (2007). Resolution strategies for fuzzy description logic. In Stepnicka, M., Novák, V., and Bodenhofer, U., editors, *EUSFLAT Conf. (2)*, pages 27–36. Universitas Ostraviensis.

[93] Hagras, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A., and Duman, H. (2004). Creating an ambient-intelligence environment using embedded agents. *Intelligent Systems, IEEE*, 19(6):12 – 20.

[94] Hailong Wang, Z. M. Ma, J. Y. (2009). FRESG: A kind of fuzzy description logic reasoner. Lecture Notes in Computer Science / Information Systems and Applications, incl. Internet/Web, and HCI, pages 443–450. Database and Expert Systems Applications: 20th International Conference, DEXA 2009, Linz, Austria, August 31 - September 4, 2009, Proceedings.

[95] Hanna Pirinen, Lotta Kauhanen, R. D.-O. J. L. I. T. N. D. R. S. S. (2015). Registered nurses' experiences with the medication administration process. *Submitted to Journal of Clinical Nursing (JCN), Wiley.*

[96] Helaoui, R., Riboni, D., and Stuckenschmidt, H. (2013). A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, UbiComp '13, pages 345–354, New York, NY, USA. ACM.

[97] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663.

[98] Hernández, A. G. and García, M. N. M. (2012). A JavaScript RDF store and application library for linked data client applications. In *Proceedings of the 21st International World Wide Web Conference, WWW2012.*

[99] Hervás, R., Bravo, J., and Fontecha, J. (2010). A context model based on ontological languages: a proposal for information visualization. *j-jucs*, 16(12):1539–1555.

[100] Hirvonen, J., Tommila, T., Pakonen, A., Carlsson, C., Fedrizzi, M., and Fullér, R. (2010). Fuzzy keyword ontology for annotating and searching event reports. In *KEOD*, pages 251–256.

[101] Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., and Devlin, S. (2009). Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing*, 5(3):236 – 252. Pervasive Health and Wellness Management.

[102] Honkola, J., Laine, H., Brown, R., and Tyrkkö, O. (2009). Smart-M3 interoperability platform release. http://sourceforge.net/projects/smart-m3/.

[103] Honkola, J., Laine, H., Brown, R., and Tyrkko, O. (2010). Smart-M3 information sharing platform. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 1041–1046.

[104] Horridge, M. (Edition 1.3, 2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools.

[105] Hosseinzadeh, S., Díaz-Rodríguez, N., Virtanen, S., and Lilius, J. (2014). A semantic security framework and a context-aware role-based access control scheme for smart spaces. *Submitted.*

[106] Hudelot, C., Atif, J., and Bloch, I. (2008). Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets Syst.*, 159(15):1929–1951.

[107] Huynh, T., Fritz, M., and Schiele, B. (2008). Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 10–19, New York, NY, USA. ACM.

[108] Ikizler-Cinbis, N. and Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 494–507, Berlin, Heidelberg. Springer-Verlag.

[109] Ismail, A.-A. and Florea, A.-M. (2013). Multimodal indoor tracking of a single elder in an AAL environment. In Berlo, A., Hallenborg, K., Rodríguez, J. M. C., Tapia, D. I., and Novais, P., editors, *Ambient Intelligence - Software and Applications*, volume 219 of *Advances in Intelligent Systems and Computing*, pages 137–145. Springer International Publishing.

[110] ISTAG (2001). Scenarios for ambient intelligence in 2010.

[111] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 559–568.

[112] J. Bailey, A. Poulovassilis, P. T. W. (2002). An event-condition-action language for XML. In *WWW '02 Proceedings of the 11th international conference on World Wide Web*.

[113] J. Feliu, J. V. and Cabré, M. T. (2002.). Ontologies: A review, Technical report IULA/INF034/02, University Pompeu Fabra, Institute for Applied Linguistics, Barcelona, Spain. Technical report.

[114] Jakkula, V., Crandall, A., and Cook, D. (2007). Knowledge discovery in entity-based smart environment resident data using temporal relations-based data mining. In *Proceedings of the ICDM Workshop on Spatial and Spatio-Temporal Data Mining*, pages 531–579.

[115] Jakkula, V., Crandall, A., and Cook, D. (2009). Enhancing anomaly detection using temporal pattern discovery. In Kameas, A. D., Callagan, V., Hagras, H., Weber, M., and Minker, W., editors, *Advanced Intelligent Environments*, pages 175–194. Springer Berlin Heidelberg.

[116] Kaustell, A., Saleemi, M. M., Rosqvist, T., Jokiniemi, J., Lilius, J., and Porres, I. (2011). Framework for Smart Space Application Development. In *Proceedings of the International Workshop on Semantic Interoperability, IWSI 2011.*

[117] Kelleher, C. (2003). Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers. *Human Factors.*

[118] Kelleher, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.*, 37(2):83–137.

[119] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems*, 3(3):263–286.

[120] Khan, N. A., Díaz Rodríguez, N., Danielsson-Ojala, R., Pirinen, H., Kauhanen, L., Salanterä, S., Majors, J., Björklund, S., Rautanen, K., Salakoski, T., Tuominen, I., Porres, I., and Lilius, J. (2014). Smart dosing: A mobile application for tracking the medication tray-filling and dispensation processes in hospital wards. In Augusto, J. C. and Wolf,

K.-H., editors, *6th International Workshop on Intelligent Environments Supporting Healthcare and Well-being (WISHWell'14). Eindhoven, The Netherlands.*, Lecture Notes in Computer Science, page 1–10. Springer.

[121] Kientz, J. A., Patel, S. N., Jones, B., Price, E., Mynatt, E. D., and Abowd, G. D. (2008). The Georgia Tech aware home. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3675–3680, New York, NY, USA. ACM.

[122] Kim, E., Helal, S., and Cook, D. (2010). Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48 –53.

[123] Kim, J. S., Kim, B. O., and Park, K. S. (2007). Development of HIHM (Home Integrated Health Monitor) for Ubiquitous Home Healthcare. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 363 –365.

[Kinect for Windows] Kinect for Windows. Developer material and instructions. *Available: http://www.microsoft.com/en-us/kinectforwindows/develop/.*

[125] Kinect@Home Project, KTH (2012). Available online: http://www.kinectathome.com/.

[126] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M., and Tran, L. (January 2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation. http://www.w3.org/TR/CCPP-struct-vocab/.

[127] Konstantopoulos, S. and Apostolikas, G. (2007). Fuzzy-DL reasoning over unknown fuzzy degrees. In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems - Volume Part II*, OTM'07, pages 1312–1318, Berlin, Heidelberg. Springer-Verlag.

[128] Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from RGB-D videos. *Int. J. Rob. Res.*, 32(8):951–970.

[129] Kowalski, R. and Sergot, M. (1986). A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95.

[130] Lai, C.-F., Huang, Y.-M., Park, J. H., and Chao, H.-C. (2010). Adaptive body posture analysis for elderly-falling detection with multisensors. *Intelligent Systems, IEEE*, 25(2):20 –30.

[131] Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view RGB-D object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE.

[132] Lassila, O. (2006). Generating rewrite rules by browsing RDF data. In *RuleML*, pages 51–57.

[133] Laszlo Kovacs, P. M. and Pataki, B. (2009). Service-oriented context-aware framework. In *Young Researchers Workshop on Service-Oriented Computing*.

[134] León, O., Cuéllar, M. P., Delgado, M., Borgne, Y. L., and Bontempi, G. (2014). Human activity recognition framework in monitored environments. In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014)*, pages 487–494.

[135] Levesque, H. J., Pirri, F., and Reiter, R. (1998). Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence*, 2:159–178.

[136] Li, F. and Dustdar, S. (2011). Incorporating unsupervised learning in activity recognition. In *Activity Context Representation*, volume WS-11-04 of *AAAI Workshops*. AAAI.

[137] Liu, C.-L. and Yang, H.-L. (2012). Applying ontology-based blog to detect information system post-development change requests conflicts. *Information Systems Frontiers*, 14(5):1019–1032.

[138] Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., and Vasudevan, V. (2009). uWave: Accelerometer-based personalized gesture recognition and its applications. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1 –9.

[139] Liu, Z. and Sarkar, S. (2006). Improved gait recognition by gait dynamics normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:2006.

[140] Lozano-Tello, A. and Gómez-Pérez, A. (2004). ONTOMETRIC : A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18.

[141] López-Matencio, P., Vales Alonso, J., González-Castaño, F., Sieiro, J., and Alcaraz, J. (2010). Ambient intelligence assistant for running sports based on k-NN classifiers. In *3rd Conference on Human System Interactions (HSI), 2010*, pages 605 –611.

[142] Lukasiewicz, T. and Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):291 – 308. Semantic Web Challenge 2006/2007.

[143] Lukowicz, P., Pirkl, G., Bannach, D., Wagner, F., Calatroni, A., Förster, K., Holleczek, T., Rossi, M., Roggen, D., Tröster, G., Doppler, J., Holzmann, C., Riener, A., Ferscha, A., and Chavarriaga, R. (2010). Recording a complex, multi modal activity data set for context recogntion. In *Workshop on Context-Systems Design, Evaluation and Optimisation at ARCS, 2010*.

[144] M. Mohsin Saleemi, J. L. (2011). End-user's service composition in Ubiquitous Computing using Smartspace approach. In *ICIW 2011, The Sixth International Conference on Internet and Web Applications and Services*.

[145] Matheus, C. J., Kokar, M. M., Baclawski, K., Letkowski, J., Call, C., Hinman, M., Salerno, J., and Boulware, D. (2005). SAWA: An Assistant for Higher-Level Fusion and Situation Awareness. In *Eighth International Conference on Information Fusion*.

[146] Maurer, U., Smailagic, A., Siewiorek, D., and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, pages 4 pp. –116.

[147] Meng, X., Lee, K. K., and Xu, Y. (2006). Human driving behavior recognition based on hidden Markov models. In *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 274 –279.

[148] Mocholí, J. B., Sala, P., Fernández-Llatas, C., and Naranjo, J. C. (2010). Ontology for modeling interaction in ambient assisted living environments. In Bamidis, P. D., Pallikarakis, N., and Magjarevic, R., editors, *XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*, volume 29 of *IFMBE Proceedings*, pages 655–658. Springer Berlin Heidelberg. 10.1007/978-3-642-13039-7_165.

[149] Moody, D. (2009). The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.*, 35(6):756–779.

[150] Morente-Molinera, J. A., Gálvez, I. J. P., Wikström, R., Herrera-Viedma, E., and Carlsson, C. (2014). Designing a decision support system

for recommending smartphones using fuzzy ontologies. In *Intelligent Systems'2014 - Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 2: Tools, Architectures, Systems, Applications*, pages 323–334.

[151] Morris, D., Schazmann, B., Wu, Y., Coyle, S., Brady, S., Hayes, J., Slater, C., Fay, C., Lau, K. T., Wallace, G., and Diamond, D. (2008). Wearable sensors for monitoring sports performance and training. In *Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on*, pages 121 –124.

[152] Murth, M. and Kühn, E. (2009). A heuristics framework for semantic subscription processing. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., and Simperl, E., editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 96–110. Springer Berlin / Heidelberg.

[153] Naeem, U. and Bigham, J. (2007). A comparison of two hidden Markov approaches to task identification in the home environment. In *Pervasive Computing and Applications. ICPCA 2007. 2nd International Conference on*, pages 383 –388.

[154] Naeem, U., Bigham, J., and Wang, J. (2007). Recognising activities of daily life using hierarchical plans. In *Proceedings of the 2nd European conference on Smart sensing and context*, EuroSSC'07, pages 175–189, Berlin, Heidelberg. Springer-Verlag.

[155] Nardi, B. (1996). *Context and Consciousness: Activity Theory and Human-Computer Interaction.* MIT Press.

[156] Okeyo, G., Chen, L., and Wang, H. (2014a). Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*, 39(0):29 – 43. Special Issue on Ubiquitous Computing and Future Communication Systems.

[157] Okeyo, G., Chen, L., Wang, H., and Sterritt, R. (2010). Ontology-enabled activity learning and model evolution in smart homes. In Yu, Z., Liscano, R., Chen, G., Zhang, D., and Zhou, X., editors, *Ubiquitous Intelligence and Computing*, volume 6406 of *Lecture Notes in Computer Science*, pages 67–82. Springer Berlin / Heidelberg. 10.1007/978-3-642-16355-5_8.

[158] Okeyo, G., Chen, L., Wang, H., and Sterritt, R. (2014b). Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10, Part B(0):155 – 172.

[159] Oppermann, R. and Specht, M. (2000). A Context-Sensitive Nomadic Exhibition Guide. In *Second Symposiumon Handheld and Ubiquitous Computing, Springer, pp. 127-142.*

[160] Pakonen, A., Tommila, T., and Hirvonen, J. (2010). A fuzzy ontology based approach for mobilising industrial plant knowledge. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8.

[161] Paulheim, H. and Pan, J. Z. (2012). Why the semantic web should become more imprecise. In *What will the Semantic Web look like 10 years from now?, Boston.*

[162] Philipose, M., Fishkin, K. P., Perkowitz, M., Patterson, D. J., Fox, D., Kautz, H., and Hahnel, D. (2004). Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57.

[163] Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990.

[164] Poveda-Villalón, M., Suárez-Figueroa, M., and Gómez-Pérez, A. (2012). Validating ontologies with OOPS! In Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., and Hernandez, N., editors, *Knowledge Engineering and Knowledge Management*, volume 7603 of *Lecture Notes in Computer Science*, pages 267–281. Springer Berlin Heidelberg.

[165] Poveda-Villalón, M., Suárez-Figueroa, M. C., García-Castro, R., and Gómez-Pérez, A. (2010). A context ontology for mobile environments. In *Proceedings of Workshop on Context, Information and Ontologies - CIAO 2010 Co-located with EKAW 2010*, volume 626, Germany. CEUR-WS.

[166] Preuveneers, D., Bergh, J. V. D., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, E., Coninx, K., and Bosschere, K. D. (2004). Towards an extensible context ontology for ambient intelligence. In *In: Proceedings of the Second European Symposium on Ambient Intelligence*, pages 148–159. Springer-Verlag.

[167] Pérez, I., Wikström, R., Mezei, J., Carlsson, C., and Herrera-Viedma, E. (2013). A new consensus model for group decision making using fuzzy ontology. *Soft Computing*, 17(9):1617–1627.

[168] Prinz, W. (1999). NESSIE: An awareness environment for cooperative settings. In *Proceedings of 6th Conference on Computer-Supported Cooperative Work.*

[169] Ramos, C., Augusto, J. C., and Shapiro, D. (2008). Ambient intelligence - the next step for artificial intelligence. *Intelligent Systems, IEEE*, 23(2):15 –18.

[170] Rantz, M., Aud, M., Alexander, G., Oliver, D., Minner, D., Skubic, M., Keller, J., He, Z., Popescu, M., Demiris, G., , and Miller, S. (2008). Tiger place: An innovative educational and research environment. In *AAAI in Eldercare: New Solutions to Old Problems.*

[171] Rashidi, P. and Cook, D. (2009). Keeping the resident in the loop: Adapting the smart home to the user. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(5):949 –959.

[172] Rashidi, P. and Cook, D. (2010). Multi home transfer learning for resident activity discovery and recognition. In *Proceedings of the International workshop on Knowledge Discovery from Sensor Data.*

[173] Reifinger, S., Wallhoff, F., Ablassmeier, M., Poitschke, T., and Rigoll, G. (2007). Static and dynamic hand-gesture recognition for augmented reality applications. *Work*, 4552:728–737.

[174] Remagnino, P. and Foresti, G. (2005). Ambient intelligence: A new multidisciplinary paradigm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):1 – 6.

[175] Remagnino, P., Hagras, H., Monekosso, N., and Velastin, S. (2005). Ambient intelligence. In Remagnino, P., Foresti, G., and Ellis, T., editors, *Ambient Intelligence*, pages 1–14. Springer New York.

[176] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and B. Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.

[177] Rex, A. (2011). Design of a caregiver programmable assistive intelligent environment. Aalto University.

[178] Riboni, D. and Bettini, C. (2011a). COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289.

[179] Riboni, D. and Bettini, C. (2011b). OWL 2 modeling and reasoning with complex human activities. *Pervasive Mob. Comput.*, 7(3):379–395.

[180] Riboni, D., Pareschi, L., Radaelli, L., and Bettini, C. (2011). Is ontology-based activity recognition really effective? In *PerCom Workshops*, pages 427–431.

[181] Rodríguez, M. A. and Bollen, J. (2007). Modeling computations in a semantic network. *Computing Research Repository (CoRR), ACM*, abs/0706.0022.

[182] Roman, M., Hess, C., Cerqueira, R., and Ranganathan, A. (2002). A middleware infrastructure for active spaces. In *IEEE Pervasive Computing.*

[183] Ros, M., Cuéllar, M., Delgado, M., and Vila, A. (2013). Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows. *Information Sciences*, 220(0):86 – 101. Online Fuzzy Machine Learning and Data Mining.

[184] Ros, M., Pegalajar, M., Delgado, M., Vila, A., Anderson, D., Keller, J., and Popescu, M. (2011). Linguistic summarization of long-term trends for understanding change in human behavior. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2080 –2087.

[185] Rowe, M., Fernandez, M., Angeletou, S., and Alani, H. (2012). Community analysis through semantic rules and role composition derivation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 0(0).

[186] Roy, N., Roy, A., and Das, S. K. (2006). Context-aware resource management in multi-inhabitant smart homes: A nash h-learning based approach. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '06, pages 148–158, Washington, DC, USA. IEEE Computer Society.

[187] Sagha, H., Calatroni, A., del R.Millan, J., Roggen, D., Troster, G., and Chavarriaga, R. (2013). Robust activity recognition combining anomaly detection and classifier retraining. In *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pages 1–6.

[188] Saguna, S., Zaslavsky, A., and Chakraborty, D. (2013). Complex activity recognition using context-driven activity theory and activity signatures. *ACM Trans. Comput.-Hum. Interact.*, 20(6):32:1–32:34.

[189] Saleemi, M., Díaz, N., Lilius, J., and Porres, I. (2011). A Framework for Context-aware Applications for Smart Spaces. In S. Balandin, S. e. a., editor, *ruSMART 2011: The 4th Conference on Smart Spaces, 2011.*, page 12, St. Petersburg. LNCS.

[190] Saleemi, M., Díaz Rodríguez, N., and Lilius, J. (2014). Erratum to: Exploiting smart spaces for interactive TV applications development. *The Journal of Supercomputing*, pages 1–1.

[191] Saleemi, M. M., Díaz Rodríguez, N., and Lilius, J. (2012a). Towards ontology-driven development of ubiquitous and intelligent interactive TV applications. In Yau, S. S., Pedrycz, W., and Piuri, V., editors, *The 10th IEEE International Conference on Pervasive, Intelligence and Computing (PiCom 2012)*, page 1–8. IEEE.

[192] Saleemi, M. M., Díaz Rodríguez, N., Suenson, E., Lilius, J., and Porres, I. (2012b). *Ontology Driven Smart Space Application Development*, page 101–125. River Publishers.

[193] Semy, S. K., Pulvermacher, M. K., Obrst, L. J., and Pulvermacher, M. K. (2004). Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation. Technical report, Workshop on Information Integration on the Web (IIWeb-04), in conjunction with VLDB-2004.

[194] Seyed, P. (2012). A Method for Evaluating Ontologies - Introducing the BFO-Rigidity Decision Tree Wizard. In *7th International Conference on Formal Ontology in Information Systems, FOIS*, pages 191–204.

[195] Simou, N., Stoilos, G., and Stamou, G. B. (2013). Storing and querying fuzzy knowledge in the semantic web using FiRE. In *URSW (LNCS Vol.)*, pages 158–176.

[196] Singla, G., Cook, D. J., and Schmitter-Edgecombe, M. (2010). Recognizing independent and joint activities among multiple residents in smart environments. *October*, 1(1):57–63.

[197] Stengel, M. (2003). Introduction to Graphical Models, Hidden Markov Models and Bayesian Networks. Tutorial, Toyohashi University of Technology, Japan .

[198] Stenudd, S. (2012). A model for using machine learning in smart environments. In *Proceedings of the 6th international conference on Grid and Pervasive Computing*, GPC'11, pages 24–33, Berlin, Heidelberg. Springer-Verlag.

[199] Stoilos, G., Simou, N., Stamou, G., and Kollias, S. (2006). Uncertainty and the semantic web. *Intelligent Systems, IEEE*, 21(5):84 –87.

[200] Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J., and Horrocks, I. (2005). The fuzzy description logic f-SHIN. International Workshop on Uncertainty Reasoning For the Semantic Web (2005).

[201] Stojanovic, L. (2004). *Methods and tools for ontology evolution.* PhD thesis, Karlsruhe Institute of Technology. http://d-nb.info/1001606787.

[202] Straccia, U. (2009). Multi-criteria decision making in fuzzy description logics: A first step. In *13th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems - KES-09*, volume 5711 of *Lecture Notes in Artificial Intelligence*, pages 79–87. Springer.

[203] Straccia, U. (2013). *Foundations of Fuzzy Logic and Semantic Web Languages.* CRC Studies in Informatics Series. Chapman & Hall.

[204] Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England.*

[205] Strang, T., Linnhoff-Popien, C., and Frank, K. (Nov 2003). CoOL: A Context Ontology Language to enable Contextual Interoperability. In *4th International Conference on Distributed Applications and Interoperable Systems*, page 236–247. IEEE Computer Society.

[206] Suárez-Figueroa, M. C., de Cea, G. A., Buil, C., Dellschaft, K., Fernández-López, M., García, A., Gómez-Pérez, A., Herrero, G., Montiel-Ponsoda, E., Sabou, M., Villazon-Terrazas, B., and Yufei, Z. (2008). NeOn methodology for building contextualized ontology networks. *NeOn Deliverable D5.4.1*, 5:4–1.

[207] Tapia, E., Intille, S., and Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, 3001:158–175.

[208] Thomas, E., Pan, J. Z., and Ren, Y. (2010). TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *the Proc. of the Extended Semantic Web Conference (ESWC2010).*

[209] Tiberghien, T., Mokhtari, M., Aloulou, H., and Biswas, J. (2012). Semantic reasoning in context-aware assistive environments to support ageing with dementia. In Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J., Hendler, J., Schreiber, G., Bernstein, A., and Blomqvist, E., editors, *The Semantic Web – ISWC 2012*, volume 7650 of *Lecture Notes in Computer Science*, pages 212–227. Springer Berlin Heidelberg.

[210] Tominaga, S., Shimosaka, M., Fukui, R., and Sato, T. (2012). A unified framework for modeling and predicting going-out behavior. In Kay, J., Lukowicz, P., Tokuda, H., Olivier, P., and Krüger, A., editors, *Pervasive*

Computing, volume 7319 of *Lecture Notes in Computer Science*, pages 73–90. Springer Berlin / Heidelberg. 10.1007/978-3-642-31205-2_5.

[211] Vail, D. L., Veloso, M. M., and Lafferty, J. D. (2007). Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 235:1–235:8, New York, NY, USA. ACM.

[212] van Kasteren, T. and Krose, B. (2007). Bayesian activity recognition in residence for elders. *IET Conference Publications*, 2007(CP531):209–212.

[213] van Sinderen, M., van Halteren, A., Wegdam, M., Meeuwissen, H., and Eertink, E. (2006). Supporting context-aware mobile applications: an infrastructure approach. *Communications Magazine, IEEE*, 44(9):96 –104.

[214] Vrandečić, D. (2010). *Ontology Evaluation*. PhD thesis, KIT Karlsruhe Institute of Technology, Karlsruhe, Germany.

[215] Walter, T. and Ebert, J. (2009). Combining DSLs and ontologies using metamodel integration. In *Proceedings of the IFIP TC 2 Working Conference on Domain-Specific Languages*, DSL '09, pages 148–169, Berlin, Heidelberg. Springer-Verlag.

[216] Wang, C., Lu, J., and Zhang, G. (2006). Integration of ontology data through learning instance matching. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 536–539.

[217] Weiser, M. (1991). The computer for the twenty-first century. *Scientific American*, 165:94–104.

[218] Wikström, R. (2014). *Fuzzy Ontology for Knowledge Mobilisation and Decision Support*. PhD thesis.

[219] Wilson, D., Wyaat, D., and Philipose, M. (2005). Using context history for data collection in the home. In *Pervasive*, volume 3468.

[220] Winograd, T. (2001). Architectures for context. *Human Computer Interaction*, 16(2):401–419.

[221] Wu, Y. and Krishnan, S. (2010). Statistical analysis of gait rhythm in patients with parkinson's disease. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(2):150 –158.

[222] X.H.Wang, D.Q. Zhang, T. G. and Pung, H. (2004). Ontology based context modeling and reasoning using OWL. In *Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications*.

[223] Xia, L., Chen, C.-c., and Aggarwal, J. K. (2011). Human detection using depth information by kinect. *Pattern Recognition*, pages 15–22.

[224] Yaouanc, J.-M. L. and Poli, J.-P. (2012). A fuzzy spatio-temporal-based approach for activity recognition. In *Proceedings of the 2012 International Conference on Advances in Conceptual Modeling*, ER'12, pages 314–323, Berlin, Heidelberg. Springer-Verlag.

[225] Yau, S. S. and Liu, J. (2006). Hierarchical situation modeling and reasoning for pervasive computing. In *Proceedings of the The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance*, SEUS-WCCIA '06, pages 5–10, Washington, DC, USA. IEEE Computer Society.

[226] Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Kontopoulos, E., Kompatsiaris, I., and Dobson, S. (2015). Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing*.

[227] Ye, J., Stevenson, G., and Dobson, S. (2014). KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, (0):–.

[228] Yin, J., Yang, Q., Shen, D., and Li, Z.-N. (2008). Activity recognition via user-trace segmentation. *ACM Trans. Sen. Netw.*, 4(4):19:1–19:34.

[229] Yun Her, Su-Kyoung Kim, Y. J. (2010). A context-aware framework using ontology for smart phone platform. In *International Journal of Digital Content Technology and its Applications, Volume 4, Number 5.*

[230] Zadeh, L. A. (1988). Fuzzy logic. *Computer*, 21(4):83–93.

[231] Zhang, D., Zhu, M., Cheng, H., Koh, Y., and Mokhtari, M. (2006). Handling heterogeneous device interaction in smart spaces. In *Proceedings of the Third international conference on Ubiquitous Intelligence and Computing*, UIC'06, pages 250–259, Berlin, Heidelberg. Springer-Verlag.

[232] Zhou, H. and Hu, H. (2008). Human motion tracking for rehabilitation, a survey. *Biomedical Signal Processing and Control*, 3(1):1–18.

[233] Zhu, Y., Nayak, N., and Roy-Chowdhury, A. (2013). Context-aware activity recognition and anomaly detection in video. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):91–101.

[234] Zimmermann, A., Lopes, N., Polleres, A., and Straccia, U. (2012). A general framework for representing, reasoning and querying with annotated semantic web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11(0):72 – 95.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www. tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

University of Granada
School of Information Technology and Telecommunications
- Department of Computer Science and Artificial Intelligence

ISBN 978-952-12-3139-1
ISSN 1239-1883

Natalia A. Díaz Rodríguez

Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces