D5.6 - Multi-Parametric Behaviour Interpretation v3

Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support

Dem@Care - FP7-288199

Deliverable Information

Project Ref. No.		FP7-288199
Project Acronym		Dem@Care
Project Full Title		Dementia Ambient Care: Multi-Sensing Monitoring for Intelligence Remote Management and Decision Support
Dissemination level:		Public
Contractual date of delivery:		Month 40, 28 February 2015
Actual date of delivery:		M41, 31 March 2015 Final Delivery: M48, 2 December 2015,
Deliverable No.		D5.6
Deliverable Title		Multi-Parametric Behaviour Interpretation v3
Type:		Report
Approval Status:		Approved
Version:		1.2
Number of pages:		90
WP:		WP5 Medical Ambient Intelligence
Task:		T5.3 Multi-parametric Patient Behaviour Interpretation
WP/Task responsible:		CERTH
Other contributors:		INRIA, UBX
Authors (Partner)		Georgios Meditskos (CERTH), Efstratios Kontopoulos (CERTH), Thanos G. Stavropoulos (CERTH), Carlos Crispim-Junior (INRIA), Francois Bremond (INRIA), Pierre-Marie Plans (UBX), Vincent Buso (UBX), Jenny Benois-Pineau (UBX)
Responsible Author	**Name**	Georgios Meditskos
	Email	**gmeditsk@iti.gr**
Internal Reviewer(s)		Eamonn Newman
EC Project Officer		Stefanos Gouvras
Abstract (for dissemination)		This deliverable reports on the third (v3) and final version of the multi-parametric behaviour interpretation framework of Dem@Care. More specifically, the deliverable presents improvements on the activity recognition algorithms from wearable videos, both in terms of scalability (execution time) and accuracy. In addition, a late-fusion scheme for event recognition has been implemented that operates on low-level data and describes the semantic information of the scene (object, place, action recognition, and people detection & tracking) to a higher level complex event recognizer. The implementation of the knowledge-driven fusion framework is described, presenting extensive evaluation results, as well as we elaborate on the final Dem@Care ontologies. Finally, the integration status of all processing components produced in this entire work package and their respective usage in pilots are presented, summarizing the research outcomes to real-world applications.

Version Log

Version	Date	Change	Author
0.1	12/02/2015	Deliverable outline	Georgios Meditskos (CERTH)
0.2	22/02/2015	INRIA's summary of contribution	Carlos F. Crispim-Junior (INRIA)
0.3	23/02/2015	UBX's summary of contribution	Vincent Buso, Jenny Benois-Pineau (UBX)
0.4	27/03/2015	UBX's contribution	Vincent Buso, Jenny Benois-Pineau (UBX)
0.5	29/03/2015	INRIA's contribution	Carlos F. Crispim-Junior, Francois Bremond (INRIA)
0.6	30/03/2015	CERTH's contribution	Georgios Meditskos (CERTH)
0.7	30/03/2015	UBX's updates	Pierre-Marie Plans, Vincent Buso, Jenny Benois-Pineau (UBX)
0.8	31/03/2015	Final draft for submission and internal reviewer comments	Georgios Meditskos (CERTH)
0.9	10/11/2015	Evaluation added to section 5	Georgios Meditskos, Efstratios Kontopoulos (CERTH)
1.0	13/11/2015	Ontology metrics added to section 6	Georgios Meditskos (CERTH)
1.1	14/11/2015	Addition of Integration and Pilot Usage section	Thanos Stavropoulos (CERTH)
1.2	18/11/2015	Final version for submission	Georgios Meditskos (CERTH)

Executive Summary

The first version (v1) of the multi-parametric interpretation framework was presented in D5.2 [21], outlining the basic methods that were adopted by the two core modules of the framework, namely the Complex Activity Recognition (CAR) and Semantic Interpretation (SI) components. More specifically, CAR serves for identifying complex activities whose modelling is grounded on information at the level of person posture and location, whereas SI addresses situations that require encapsulating pieces of information of higher abstraction.

The second version (v2) of the multi-parametric behaviour interpretation framework was presented in D5.4 [45], reviewing state-of-the-art approaches relevant to the interpretation objectives of WP5 and describing extensions to v1 for supporting reasoning under uncertainty and handling incomplete and noisy input. The report also elaborated on functional extensions to v1, such as the support of questionnaire-related data and the incorporation of a Complex Event Processing engine to provide basic real-time interpretation services.

This document reports on the third (v3) and final version of the multi-parametric behaviour interpretation framework. More specifically, the deliverable presents improvements on the activity recognition algorithms from wearable videos, both in terms of scalability (execution time) and accuracy. In addition, a late-fusion scheme for event recognition has been implemented that operates on low-level data and describes the semantic information of the scene (object, place, action recognition, and people detection & tracking) to a higher level complex event recognizer. Finally, the implementation of the knowledge-driven semantic interpretation framework is described, presenting extensive evaluation results, along with the final version of the Dem@Care ontologies.

Abbreviations and Acronyms

ADL	Activities of Daily Life
AO	Active Object
VP	Visual Place
CAR	Complex Activity Recognition
DLs	Description Logics
DnS	Descriptions and Situations
DTI-2	Philips Discrete Tensions Indicator
DUL	DOLCE UltraLite
Gear4	Gear4 Renew Sleep Clock
HAR	Human Action Recognition from static and wearable cameras
IADL	Instrumental Activities of Daily Living
KB	Knowledge Base
KBM	Knowledge Base Manager
ORWC	Object Recognition from Wearable Camera
OSA	Offline Speech Analyser
OWL	Ontology Web Language
OWL-DL	Ontology Web Language Description Language
PDT-PER	People Detection, Tracking and Primitive Events Recognition
PwD	Person with Dementia
RDF	Resource Definition Framework
RRWC	Room Recognition from Wearable Camera
SI	Semantic Interpretation
SPARQL	SPARQL Protocol And RDF Query Language
SPIN	SPARQL Inferencing Notation
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
TURTLE	Terse RDF Triple Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WCPU	Wearable Camera Processing Using
WP	Work Package
XML	eXtensible Markup Language

Table of Contents

List of Figures

List of Tables

1 Introduction

The goal of multi-parametric behaviour interpretation in the Dem@Care project is to recognise the behaviour of the person with dementia (PwD) and discern traits that have been identified by the clinicians as relevant for diagnostic, status assessment, enablement and safety purposes. To this end, the information made available by WP3 and WP4 regarding physiological and lifestyle characteristics, as well as information regarding activities of daily living, is fused and aggregated in WP5 to derive high-level interpretations and decision support tasks.

In order to implement multi-parametric behaviour interpretation, two constituents need to be considered for supporting the underlying fusion tasks, namely representation and interpretation. The representation layer in Dem@Care provides the ontology vocabulary and infrastructure for capturing and storing information relevant to the lab and home/nursing home environments, such as:

- Atomic activities and measurements detected by means of WP3/WP4 monitoring and analysis components (e.g. speech events, body temperature, light level, etc.), and complex activities inferred by WP5 CAR and SI components (e.g. having meal, sleeping, napping, answering the phone, having a face-to-face conversation, etc.).
- Problems and situations that the clinicians need to be informed about (e.g. missed meals, excessive napping, insufficient communication attempts, nocturia, etc.).
- Clinically relevant attributes and summaries (e.g. sleep efficiency and duration, number of daily telephone and face-to-face interactions, night sleep summaries, etc.).

The representation models have been initially defined in D5.1 [20] and several revisions have been made since then to ensure that the representation layer adequately covers the knowledge that it is expected to capture through descriptive, yet lightweight ontology models.

The interpretation functionality of Dem@Care has been incrementally implemented in three prototypes (versions), with each version tackling different challenges. The first prototype has provided the modules for preliminary complex activity recognition and high-level aggregated interpretation of the observations captured by WP3/WP4 monitoring services. In the second prototype, the focus has been on the development of methodologies for handling uncertainty, incomplete and noisy input, as well as for supporting patient-tailored feedback services.

In the third and final prototype, the emphasis has been placed on improving the performance and scalability of the framework, as well as on enriching the framework with novel fusion mechanisms. More specifically, regarding activity recognition from wearable camera, we developed an approach that enables the transformation of instantaneous activity recognition output into a series of sequential semantic events (Section 3.1). This approach has been used for contextual reasoning in Home scenarios towards assessing the patient behaviour. In addition, we present evaluations of computation times of all processing chains in activity recognition, such as recognition per se (testing), learning phase (training) and data preparation (annotation) for wearable video sensors (Section 3.2). Finally, we present a study on how multimodal contextual reasoning can profit from the integration of wearable sensor data, presenting our efforts to exploit the DTI wrist sensor for improving the activity recognition from wearable video sensors (Section 3.3).

Towards enriching the fusion capabilities of our framework, we propose a hierarchical framework for high-level multimedia event recognition (Section 4). The framework adopts a late-fusion scheme, where intermediate event detectors (also known as concepts) operate on low-level data and describe the semantic information of the scene (object, place, action recognition, and people detection & tracking) to a higher-level complex event recognizer. These heterogeneous sources of primitive states and events are first temporally synchronized using semantic information, and then incorporated as components of event models hand-crafted using an ontology language. Event inference works using a temporal algorithm adapted to optimize event recognition based on semantic information about the scene, event probability and sensor reliability. The proposed framework performance is demonstrated on event recognition using recordings of older people undertaking daily living activities recorded by standard colour, colour-depth and wearable sensors.

Additionally, we describe a proof of concept implementation of the knowledge-interpretation framework presented in D5.4 (Section 5). More specifically, we describe the new ontologies that can be used for modelling information in various levels of abstraction and we elaborate on the architecture and technologies that underpin the implementation. The key idea of the framework is to use ontologies for representing dependencies among high-level situations and low-level observations in a loosely-coupled manner, rather than defining strict contextual patterns that cannot provide enough flexibility for handling the imprecise and ambiguous nature of real-world events. The contextual information encapsulated inside the dependency models is used for identifying links among observations that signify the presence of complex activities and to subsequently classify them as high-level activities. In addition, we present extensive evaluation results of the framework on nine common ADLs (activities of daily living). Section 6 elaborates on the final Dem@Care ontologies, presenting this year's standardisation efforts, evaluation results using the OOPS! tool and relevant metrics.

Finally, the deliverable presents a component integration and pilot usage section, which summarizes the contributions of this entire work package to real-world piloting, effectively linking research outcomes to practical, clinical usage.

2 Final Architecture and Achievements Overview

Figure 1 presents the interactions of the WP5 components in the final version of the framework. More specifically, the CAR component detects activities from RGB-D video streams. The detected activities are sent to the CEP module for real-time fusion with profile knowledge. The CEP engine queries the KB to retrieve behaviour patterns and the events that are detected are sent to the alert and feedback services of WP6. Note that the activities sent by CAR are also stored in the KB for further offline processing and fusion with other observations by SI. In parallel, the WCPU component detects activities from wearable camera by fusing objects, places, and motion information acquired from the DTI2 sensor. All the detected activities are stored in the KB, as well as in the relational database maintained locally by the CAR component, so as to apply the offline fusion algorithms (see section 4).

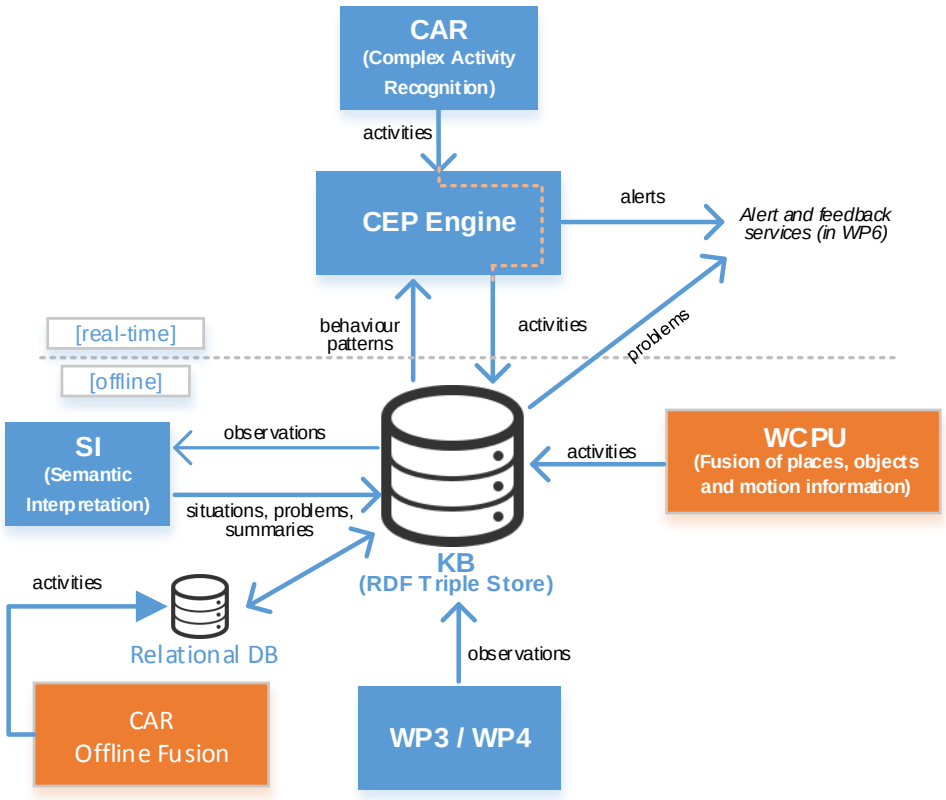


Figure 1 Logical component architecture within WP5

An overview of the key results and achievements reached within WP5 is given below:

- Multi-sensor activity recognition
  - Fusion of person tracking and zones (RGB-D / 2D camera)
  - Fusion of objects and places
  - Uncertainly handling
- Detection of problematic situations
  - Lab (e.g. missed activities)

    - Home/Nursing Home (e.g. sleep problems, reoccurring problems)
- Activity-oriented detection of physical-related measurements (RGB-D data)
    - Walking speed, stride length, etc.
- Extraction of high-level patient-tailored norms/patterns
    - Duration/frequency of certain activities
    - Deviations from normal behavior
- Correlations/contributing factors
    - e.g. correlations between physical activity and sleep problems (such as, when low physical activity affects the quality of sleep)
- A number of ontologies have been defined and published for formally capturing activity and profile knowledge
    - Lab, Home and Nursing home settings
    - Reusing and extending existing upper-level ontologies
    - Reusable/shareable models
- Questionnaires
    - Different questionnaire types are supported for sleep, mood, etc.
- Daily summaries of activities
    - Aggregation of information into knowledge structures that are used in WP6 for presentation purposes

3 Event Recognition from Wearable Video Cameras and Sensor Fusion

In this section we describe the improvements that have been implemented in v3 regarding activity recognition from wearable camera. More specifically, we present our approach in transforming instantaneous (per-frame) activity recognition output into a series of sequential semantic events. In addition, we present evaluation results of the computation times of all processing chains in activity recognition, such as recognition per se (testing), learning phase (training) and data preparation (annotation). Finally, we present a study on how multimodal contextual reasoning can profit from the integration of wearable sensor data and we elaborate our efforts to exploit the DTI wrist sensor for improving the activity recognition from wearable video sensors.

3.1 Event-based Format

Action recognition results are currently widely used as per-frame results in which each frame has their own recognition probabilities for each action class (e.g. using a phone or reading a newspaper). This kind of result is useful in order to work with other devices, as it is the raw data format, and thus we have no loss of information. Nevertheless, this kind of format is not easily readable, since much probability information is included in the files. In addition, the files taken alone do not provide class information, so it is difficult to know what information a probability value is related to. In order to avoid such kind of problems, we are investigating a more human-readable format that will help us quickly determine the predominant event at a specific time and to what class it is related. In order to obtain that kind of format we have to transform the per-frame format with loss of information, using the algorithm described in the following section.

We use XML to represent the events, where the nodes are composed of four text/data elements: two refer to time information (event begin and end), one describes the category of the event and the last one provides the probability of this event to be the correct one. The last two elements are embedded inside a “concept” element and can be repeated within the “concepts” element. This format allows easy exchange of information with other modules developed within the Dem@Care project and it is mainly used in the fusion framework described in Section 4. An example of the XML format used for representing events is given below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<videoObservation>
	<source>resultFile</source>
	<videoSegmentObservations>
		<begin>2013-12-17T03:54:57.033+01:00</begin>
		<end>2013-12-17T03:55:17.000+01:00</end>
		<concepts>
			<concept>Preparing_drugs</concept>
			<probability>0.999356</probability>
		</concepts>
	</videoSegmentObservations>
	<videoSegmentObservations>
		<begin>2013-12-17T03:55:17.033+01:00</begin>
		<end>2013-12-17T03:55:21.000+01:00</end>
		<concepts>
```

```
                    <concept>Reading_instructions</concept>
                    <probability>0.417044</probability>
                </concepts>
            </videoSegmentObservations>
</videoObservation>
```

In order to get per-event result files instead of per-frame results, we have developed a two-phase algorithm that transforms the results of the activity recognition module to the aforementioned XML format. Initially, we transform the result files into tree-structured data, as depicted in Algorithm 1a.

Algorithm 1a

Inputs

resultFile: the ORWC, RRWC or ARWC result file

catFile: the categories file corresponding to the results

date: the date corresponding to the video recording date time

transform(resultFile, catFile, date)

begin

observations<-results_to_objects(resultFile);

windowSize<-10;

temporalObservations<-temporalSmoother(observations, windowSize);

print(eventsAsXML(temporalObservations));

end

Then, we find the most probable event that will enable the merging of the right result frames with the associated "Most probable" event, as described in Algorithm 1b. Once the process is complete, we just transform the structured data in memory into XML data format.

findMostProbableEvent (MPE) gives the event with the highest probability value from the current observation to the observation at *t+windowSize*. Given O the overall observations, $O(t)$ the observation at a time t for every concept, O_c the observations with the concept c and $O_c(t)$ the observation with the concept c in a time t, the formula that gives the most probable event in a window T is:

$$MPE(O(i), w) = MAX(for\ each\ c: \sum_{t=i}^{w} O_c(t))$$

The final result is an XML file that gives us the information of beginning and end of an event based on the results obtained by our Activity recognition method.

3.2 Scalability through Execution Time Measurement

In this section, we present the scalability results we obtained by testing the modules for activity recognition (ARWC), object recognition (ORWC) and place recognition (RRWC). The study has been performed within UBX using two corpora from DCU and CHU Nice.

Algorithm 1b

Inputs

observations : the ORWC, RRWC or ARWC results as Object tree

windowSize : the window size used to find the most probable event

event : {begin : date, end : date, concept : string };

transform(observations, windowSize)

begin

```
events<-emptyList();
currentEvent<-event();
for each observation  in observations-windowSize
do
        mpe<-findMostProbableEventIn(observation, windowSize);
        if(isEmpty(currentEvent())
        then    currentEvent<-mpe;
                setEventBegin(currentEvent ,observation.getBegin());
        else if (mpe!=currentEvent)
                setEventEnd(currentEvent, observation.getEnd());
                currentEvent<-mpe;
                setEventBegin(currentEvent, observation.getBegin());
                events->add(currentEvent);
        endif
endfor
if(isNotEmpty(currentEvent))
then    setEventEnd(currentEvent, observation.getEnd());
        events->add(currentEvent);
endif
return events;
```

end

Each dataset has different categories and number of videos, which explains the differences in execution speed. In order to do this study, we launched the analysis on a certain amount of videos and we registered the elapsed time between the moment we started the analysis and the moment it finished. The results are depicted in Table 1.

Table 1 RRWC, ORWC and ARWC execution times

Computational Time RRWC				
	No. Frames	**No. Categories**	**Total duration**	**Time / Frame**
DCU	4546	13	0h15	0.2041487022s
CHUN	1798	8	0h11	0.3789599555s
Computational Time ORWC				
	No. Frames	**No. Categories**	**Total duration**	**Time / Frame**
DCU	4546	16	2h01	1.6065266168s
CHUN	1798	21	1h10	2.3590433815s
Computational Time ARWC				
	No. Frames	**No. Categories**	**Total duration**	**Time / Frame**
DCU	4546	17	2h18	1.8343004839s
CHUN	1798	10	1h22	2.7502002225s

The Time / Frame column depicts the seconds the algorithm needs to analyse a frame and is greatly affected by the number of frames and number of categories. Ideally, we want to reach the real-time execution, so as to compute at least 25 frames per second. However, we are still far from this. It should be noted thought that the focus on our research in activity recognition from wearable cameras has been mainly on improving the accuracy performance rather than achieving real-time execution. Nevertheless, efforts have been made towards improving the execution time, with several parts of our components (e.g. saliency computing for object recognition) having been implemented using GPU processing technologies and optimizations.

3.3 Fusion of video and 3D motion tracking data for recognition of instrumental activities from the wearable video sensor

So far, our research in activity recognition from wearable camera has mainly focused on video/image processing techniques, i.e. object and room recognition, and activity recognition through fusing objects and places. The results we have obtained and described in previous deliverables were satisfying but there is still potential to improve them. Motivated by this fact, we propose a new method that enriches the activity recognition task by using motion tracking sensor data. More specifically, we present in this section our approach in using motion tracking information in order to further enrich the input already provided to our activity recognition algorithms, based on the following assumptions:

- Active objects, either manipulated or observed by the user, provide very strong cues about the action
- Context also contributes to the active objects by identifying the place in which the action is being made
- Energy/movement information provided by a wearable device can further provide useful insight into the ongoing activity.

3.3.1 DTI2 and Dataset

DTI2 is a wearable bracelet (Philips, WP3) able to detect movement energy. The bracelet is actually worn by patients in the hospitals during the experimentations, so that we have movement tracking data for all the experiments to build an action recognition system.

We have implemented our models using a Lab dataset (CHUN) with 21 videos for training and 23 videos for testing. We used the same dataset with and without DTI data, so as to compare the results obtained. Each patient is wearing DTI2 on the left arm and a GoPro camera on the right shoulder focused on the hands of the patient. The patients have a set of pre-defined activities to perform in a small room without a specific order. This setting makes the video analysis very challenging because the environment is different for each activity and the patient is moving in the room. The videos were recorded on the h264 norm with a resolution of 1280x960 and 30 frames per second.

3.3.2 Description of the method

The proposed activity recognition algorithm implements a hierarchical approach with two connected processing layers. The first one contains a set of object detectors (ORWC outputs), place detectors (RRWC output) and energy values (DTI outputs). The second layer uses the outputs of the first layer to perform the activity recognition task. The full pipeline is depicted in Figure 2:

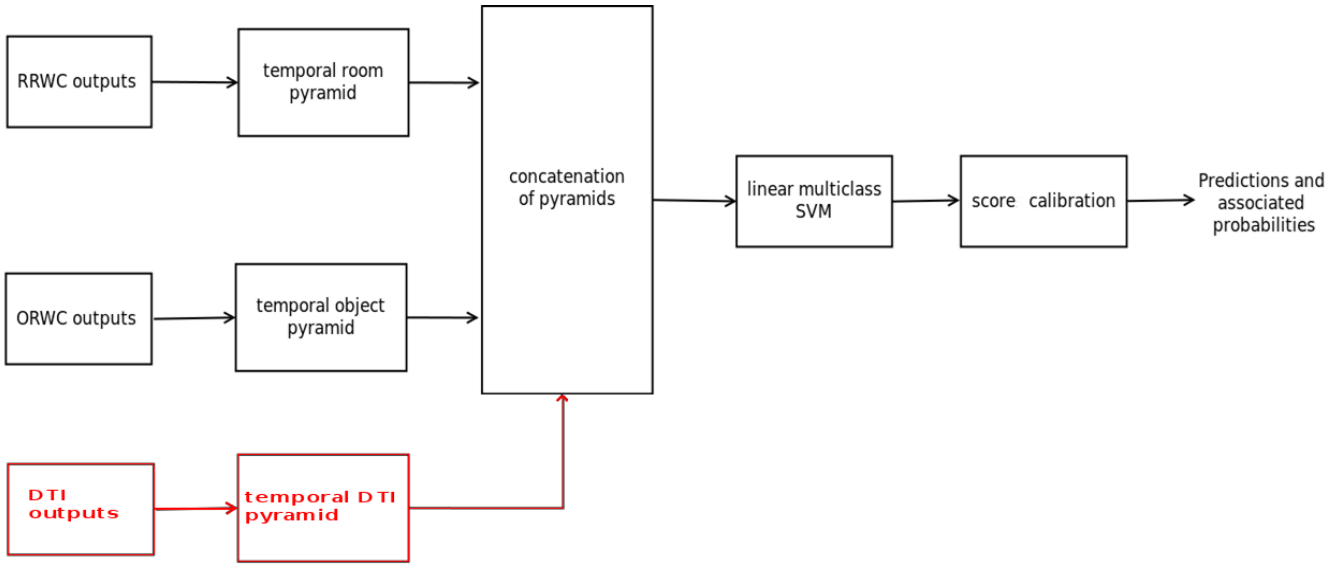


Figure 2 Activity recognition by fusing location, objects and movement energy

The overall activity recognition process follows the same scheme described in D5.5. In addition, we added the DTI outputs as energy values input to the process in order to improve the activity recognition performance. Before adding the DTI outputs, we have to pre-process them in order to obtain energy values that can fit the pipeline.

Synchronizing DTI outputs

In order to process video and DTI data, we first need to ensure that these data is synchronized. DTI is providing data in text format containing lines in the following syntax (see Table 2):

yyyymmddHHMMSSmmm|NNNN|ev|ttttt|sksks|xxxxx|yyyyy|zzzzz|vb|etete|eieie|rrrrr|cccccccc

Table 2 DTI text format specification

Layout code	Description	width	Format
yyyy	Year	4	201x
mm	Month	2	01-12
dd	Day	2	01-31
HH	Hours	2	00-24
MM	Minutes	2	00-60
SS	Seconds	2	00-60
mmm	milli seconds	3	000-999
NNNN	Serial number	4	0001-9999
ev	Event	2	1=button; 2=detected
ttttt	Skin Temperature	5	100x oC
sksks	Skin Conductance	5	nano Siemens
xxxxx|yyyyy|zzzzz	3D acceleration (x,y,z)	15	Counts (16384 = 1g)
vb	Battery voltage	2	10x Voltage
etete	Ambient temperature	5	100x oC
eieie	Ambient light level	5	Counts
rrrrr	Skin conductance ADC	5	16bit raw ADC output
cccccccc	Counter	8	Counts

Preliminary studies resulted in two ways of synchronizing DTI data and GoPro videos. The first one was based on the visual inspection of the video, identifying movements on the hand that were also reflected in DTI2 values. The second way was found after receiving a new set of videos from DCU. In these videos, we observed that the patient was stopping and then starting DTI2 after a few seconds. Therefore, we could synchronize the data by simply locating the same behavior on the video and computing the temporal difference. This works well, but unfortunately the set of videos provided was not usable since we had no recognition models pertinent to his set.

Since the synchronization could not be performed using data from home settings, we base our work on CHUN data (Lab), since we had both a recognition model ready for these videos and the corresponding DTI2 data files (Figure 3, Figure 4, Figure 5). However, we still faced some problems using this data:

- Some videos did not have the correct DTI files
- Some videos did not have DTI files with the correct timestamps corresponding to our GoPro time synchronization.

In order to perform the synchronization, we developed a tool that shows the three curves (one for each x, y and z values) for a file. Using this tool, we found the start and end DTI frames of periods of inactivity that can be recognized by a plateau with noisy values and then, with another tool, we retrieved the times corresponding to these frames. With that time span and

the start frame time we were able to find the corresponding non-activity time in the video. For the synchronization, we only used the y-axis values from DTI. This value seemed to be the only one we could trust because our camera is attached on the patients' right shoulder so that the only movement easily noticeable is the vertical movement.

Once we identified the beginning of the video recordings, we began to crop DTI2 files in order to fit each video with the right DTI2 values. Usually we only need one or two videos from the 3 or 4 available in each experiment because we are only interested in videos where the participant is alone, without the clinician.

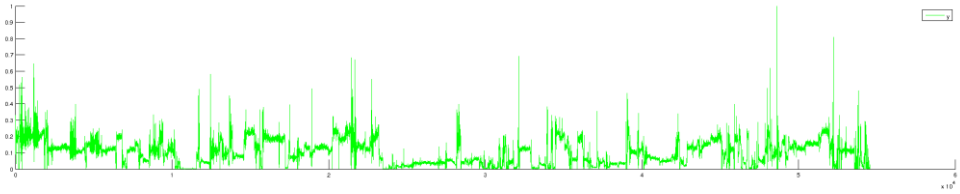


Figure 3. DTI2 overall dataset for Y-Axis

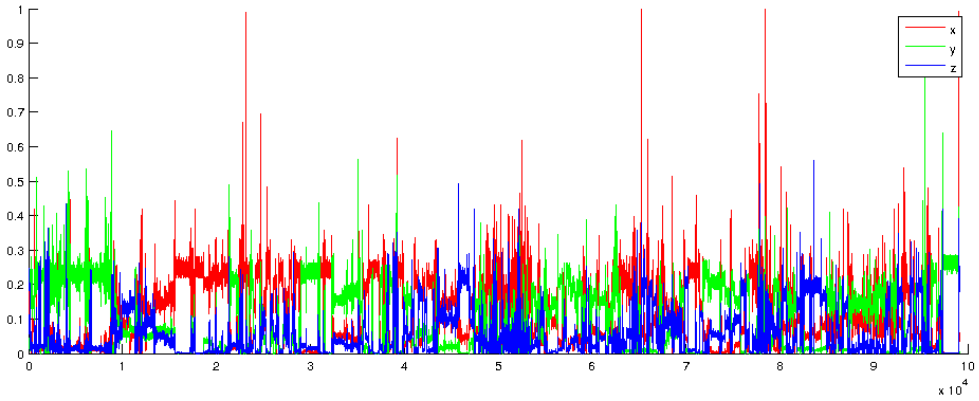


Figure 4. DTI2 overall dataset 3-axis

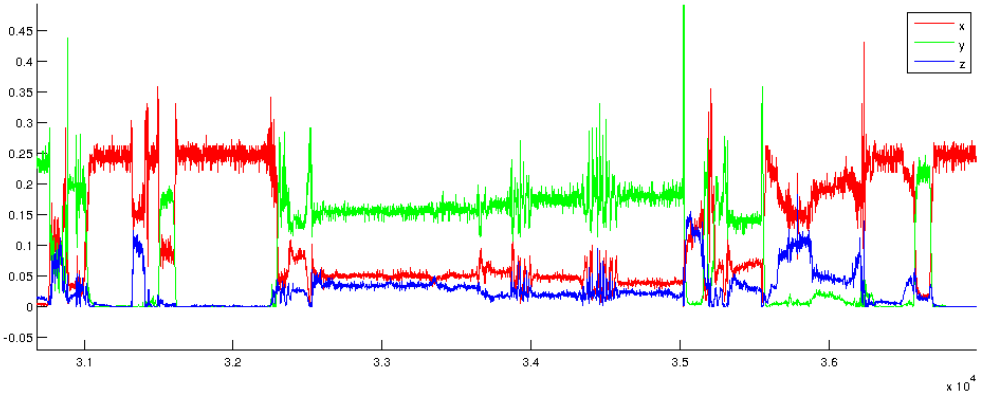


Figure 5. DTI2 dataset from Figure 7 in [3.1;3.6]

Computing DTI energy

In order to obtain energy (ε) values from the DTI outputs, we consider that all DTI outputs are three-dimensional data that we call α. For each component of α we know its maximum value denoted as MAX_x, MAX_y and MAX_z. First, our pre-process pipeline has to normalize the DTI values. Then, we transform them in order to obtain only positive energy values, normalizing the DTI raw data as:

$$normalize(a) = \begin{pmatrix} \frac{a_x}{MAX_x} \\ \frac{a_y}{MAX_y} \\ \frac{a_z}{MAX_z} \end{pmatrix}$$

Once we obtain the normalized α, we compute ε for every "second", in which T is the number of α in a second. The actual average data rate T value is 25. Our method consists to square all the normalized values. This done, we obtain the real activity information. We used this formula:

$$\varepsilon = \frac{1}{T} \sum_{t=1}^{T} \begin{pmatrix} a(t)_x^2 \\ a(t)_y^2 \\ a(t)_z^2 \end{pmatrix}$$

For the last formula, because we have no certitude about the data rate, we focus on time information given by DTI2. With the synchronization done, we know the exact time when the recording began. We get the DTI2 row with the nearest value and then we compute the energy ε by adding the subsequent rows until a second's worth of data has been computed. Now we have the DTI2 values corresponding to the video with the best possible accuracy, since the T (data rate) value is not really trustworthy.

3.3.3 Discussion of the results

Once the synchronization was completed, we integrated the DTI2 data into ARWC obtaining interesting results, capitalising on the hypothesis that the motion information from a motion tracker could confirm our recognized activities.

We have performed a comparison between ARWC without motion data and ARWC enriched with DTI2. The second method achieved better results in activities that need a certain amount of movement. We computed the mean accuracy for the results in all the 11 categories of the dataset: 67.4% for ARWC and 69.4% for ARWC+DTI2. We have chosen to highlight five of these categories that were improved by the integration of DTI2. The mean accuracy for ARWC is about 64% for these 5 categories and 66.5% for ARWC+DTI2 (Figure 6). We can observe an improvement of 2% for the overall categories and 2.5% for the categories that received an improvement with our new recognition method. For the 5 activity categories, the variance for ARWC is about 0.068 where in our new method is 0.056, while the standard deviation for ARWC is 0.256 and 0.238 for our method with DTI2 fusion. This means that our new method is more precise and the values are less spaced with this method (Figure 7).

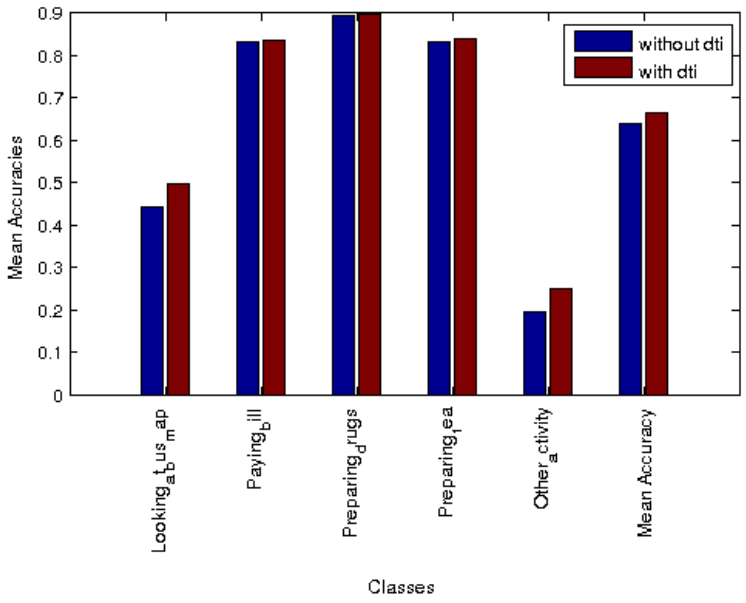


Figure 6. Comparison between without motion tracking results and our first method involving motion tracking

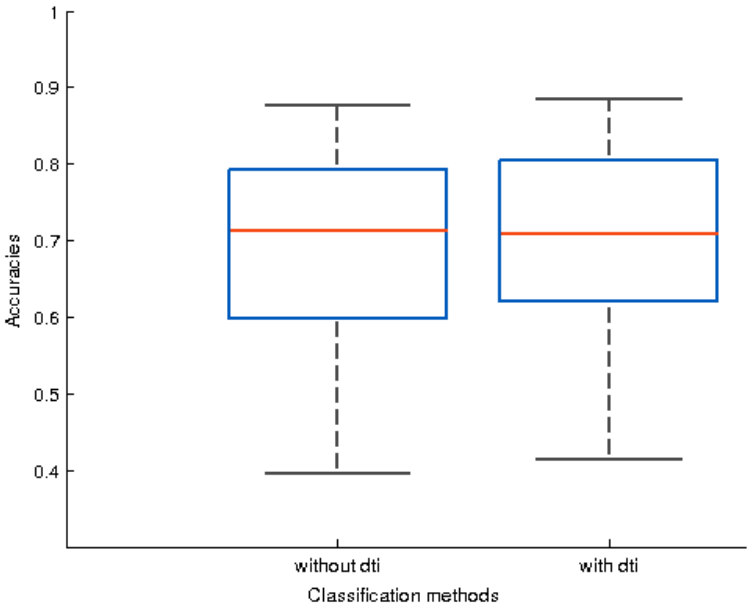


Figure 7. Comparison between without motion tracking method and our first method involving motion tracking

All in all, this study has shown that a fusion between video analysis and motion tracking analysis can improve the results of activity recognition. As expected, only the activities that require noticeable amount of movement have been improved. As far as the implementation is concerned, several improvements need to be incorporated regarding the synchronization, which is currently performed manually. The methods used to process the DTI2 results can be also improved by more sophisticated transformations.

4 Semantic Event Fusion of Different Visual Modality Concepts for Activities of Daily Living Recognition

4.1 Introduction

The analysis of multimodal data for event recognition has gained focus recently, especially after the popularization of consumer platforms for video-content sharing such as YouTube and Vimeo. The need to automatically analyze and retrieve subsets of video content according to textual or image queries has motivated research about the ways visual content can be semantically described.

This work focuses on a similar but different problem: multimodal monitoring of peoples' activities in an ecological scenario, where people undertake natural activities of daily living. This problem is in a more constrained setting than YouTube videos, as we take advantage of 3D scene information since changes to sensor position are kept to a minimum. However, other issues arise in daily life monitoring and challenging requirements exist. For instance, it is necessary to accurately detect people, to fuse heterogeneous sensor data with different frame rates, to recognize multiple low-level events within a single recording and across different modalities, and to handle data incompleteness due to the variable field of view (FoV) of the different sensors. Finally, this line of research also targets a reliable assessment of temporal boundaries of an event interval. This last constraint is not commonly addressed in multimedia event recognition, since most research focuses on image/video content retrieval, but is very pertinent in domains like medical research, as automatically obtained data is used to derive new statistical indicators about the patient health status. Multimodal video content retrieval field has investigated ways to represent video content by associating different image cues, like motion, and appearance, as in [42], to other modalities commonly present in video recordings, such as audio and text. In [34], the authors have introduced a feature-level representation that combines audio and video features by modeling the joint patterns between these modalities during each event class of interest. Oh *et al.* [51] have proposed a multimodal (audio and video) event recognition system, where base classifiers are firstly learned from different subsets of low-level features, and then combined with mid-level features, such as object detectors, to recognize complex events. In similar settings, Myers *et al.* [47] have showed average output is a simple but very effective method to perform late fusion for event recognition on a set of classifiers, each learned from a single data type/ source -- *e.g.*, low-level vision, motion, audio, but all targeting the same events. Pinquier *et al.* [52] proposed an intermediate fusion approach based on a hierarchical Hidden Markov Model for the recognition of activities of daily living from wearable camera data (video and audio).

These data-driven approaches explore different levels of data abstraction, in order to decompose complex events into intermediate representations, such as object detectors, which try to better capture event semantics and handle the naturally hierarchical structure of high-level events. Although these are significant advances, most of the described methods require large amounts of training data to learn event semantics and achieve a proper generalization. Moreover, these approaches focus on classifying an entire video clip, whereas in our setup it is necessary to detect spatiotemporal regions of the video where activities occur and classify them.

Research focusing on daily living activity monitoring has been proposed by the Ambient Assisted Living community, and is generally based on environmental sensors, such as passive

infrared, contact sensors, RFID tags [25][46]. Chen *et al.* [15] have proposed a hybrid approach between knowledge-driven methods (ontology-based) and data-driven approaches for activity modeling and recognition based on a miniaturized sensor attached to objects of daily living. Domain heuristics and prior knowledge are added to event models as seeds for initialization, and then data-driven methods are used iteratively to update the profile of the monitored user (patient profile). Cao *et al.* [12] have proposed a multimodal event recognition approach, where context is employed to model the human and the environmental information. Human context (*e.g.*, body posture) is obtained from cameras, while the environmental context (semantic information about the scene) is based on accelerometers attached to objects of daily living. A rule-based reasoning engine is used to include context in complex event recognition.

Knowledge-driven methods and its variants have recently gained attention [12][18][15], as they facilitate the incorporation of prior knowledge to event models, the definition of rules and constraints over modeled concepts by incorporating logic and interpreting model failures. At the same time, such approaches are sensitive to both the performance and the presence of noise in the observations provided by the underlying components. Probabilistic variants of these semantic approaches have been investigated [61], in order to handle data uncertainty at the same time it take advantage of explicit knowledge about the domain of application events and the scene characteristics.

In this work, we propose a novel probabilistic knowledge-driven multimedia framework for event recognition. We use data-driven methods to learn low-level concepts, a hierarchical knowledge-driven approach to link concepts to event models, and a semi-probabilistic inference algorithm to recognize complex events based on the satisfaction of their knowledge-driven constraints and the probability of their concepts. The probabilistic algorithm allow us to explore the complementarities and discrepancies of different types of visual data by handling noisy observations, disagreement between competing sources of information, incomplete evidence, and the variance in concept recognition performance across sensors. We also present a novel approach to tackle the problem of temporal alignment of sensor data streams at semantic level, since even though sensor raw data are assumed to be coarsely synchronized, time shifts might exist between the real-world phenomena and the recognized concepts, *e.g.*, due to different and variable frame rate of sensors, and from artifacts introduced by the algorithm used for conceptual data extraction. Finally, we also introduce the combined usage of complementary visual sensing modalities for daily living activity recognition, by deploying wearable cameras, 2D static and color-depth cameras.

This chapter is organized as follows: Section 4.2 describes our multimodal approach for concept recognition from heterogeneous visual sensors; Section 4.3 presents our framework for multimedia event representation and recognition; Section 4.4 presents the dataset and the baseline methods used to evaluate the performance of the framework; and Sections 4.5 and 4.6 present Results and Discussion, respectively.

4.2 Multimedia Concept Recognition

The multimedia event recognition framework is structured in a hierarchical fashion where, firstly, we use a set of detectors to recognize low-level concepts from raw sensor data, then we link detected concepts to the conceptual world using an ontology language representation of the multimedia event hierarchy as physical objects and low-level events, and lastly we perform event inference following a knowledge-driven representation and a semi-probabilistic

inference method. The visual concept detectors of the multimedia framework adopt the following vocabulary:

- **Concept**: any object from the real world or derived from it that is modeled as a physical object in the ontology language.
- **Detector**: a process that provides an interpretation of raw sensor data to the conceptual world.
- **Instance**: a given example of a concept.

Four types of detectors are used for this framework and they are based on three different visual sensors: a 2D static camera, a color-depth static camera, and a wearable camera. The 2D camera data is used for action detection in videos of long duration and subsequent action recognition (subsection 4.2.1). The wearable camera data is used to derive information about the objects being handled and the location (place) where the person is from a first-person perspective of the scene (subsection 4.2.2). Finally, the color-depth data is used to detect people, delimit their body with a bounding box, identify their posture, and localize them over space and time (subsection 4.2.3). Each visual modality is individually processed to obtain a set of concepts from their point of view of the scene.

Figure 8 illustrates the multimedia event recognition system architecture. The detectors (A-D) process their input sensor data (*S0-S2*) and provide their results as an intermediate representation for high-level event inference. The multimedia event Recognition uses this conceptual representation to build low-level event models and then infer more complex activities from the composite and temporal relationship between events.

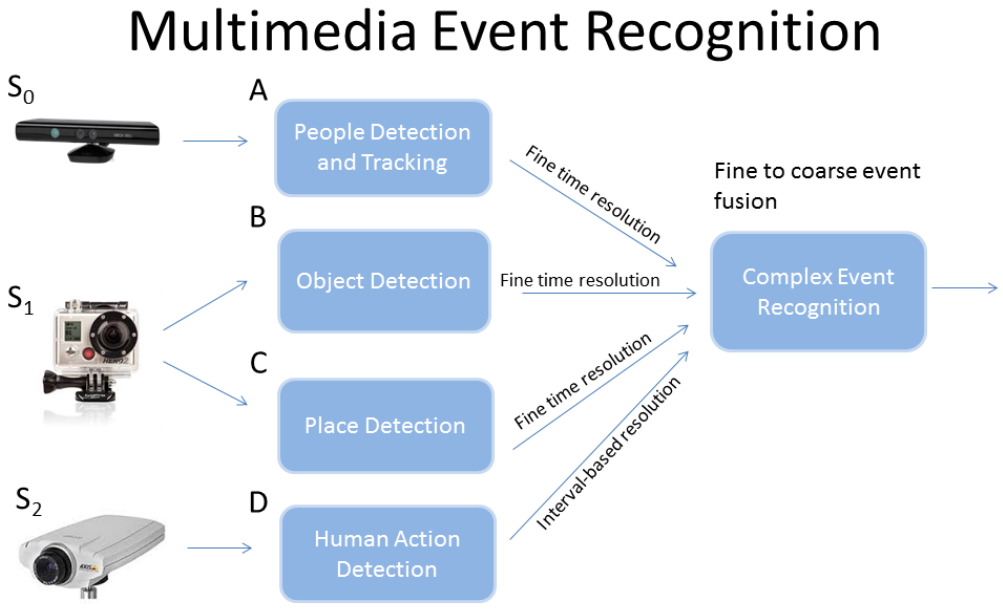


Figure 8. Conceptual architecture of the Multimedia Event Recognition System.

4.2.1 Action detection from color images

Action detection is usually addressed in the State-of-the-Art by localizing actions using a sliding spatio-temporal window [37]. However, these approaches entail a high computational cost and actions are localized in rectangular spatial areas, corresponding to the time window used. We propose a novel algorithm for spatiotemporal localization that overcomes the limitations of the current window based state of the art methods by reducing the

computational cost while achieving higher accuracy. Spatial localization occurs in regions of changing motion, so-called Motion Boundary Activity Areas (MBAAs). Temporal localization deploys statistical change detection, applied sequentially to the outcomes of a SVDD-based classifier at each frame, for Sequential Statistical Boundary Detection (SSBD). This results in a faster activity boundary detector, which is based on the statistical properties of the training and testing data. Action cuboids are then extracted in the resulting subsequences and their motion and appearance properties are used for recognition in a multiclass SVM model. The main novelty of our approach lies in the spatio-temporal activity localization, which is performed on a binary classification level and then is extended to multiple classes, in order to detect the activity that occurs in a specific temporal interval. Spatial localization also takes place in an original manner, by isolating regions of changing activity, thus avoiding false alarms and increasing the system's accuracy.

The proposed method is theoretically founded and makes intuitive sense, as it is based on finding changes in the statistical characteristics of the activities taking place. It goes beyond the SoA, as it avoids the use of costly spatiotemporal sliding windows, leading to improved computational efficiency and higher classification accuracy. More details about this detector's techniques are available in [3][4].

Action detection provides valuable cues on about actions given their motion patterns, as it uses a global representation of image/video segment, but they generally do not identify the author(s) of the action. For such reason, its usage in the multimedia framework is a natural complement for people detection and tracking modules and derived events (subsection 4.2.3).

4.2.2 Concepts from Egocentric vision

Static cameras/sensors may suffer from occlusion and clutter in unconstrained scenes, a situation that only deteriorates as the person moves away from the camera, both for action detection and people detection and tracking approaches. For this reason, in addition to static color video data, wearable video recordings are also employed, as egocentric vision provides two additional visual cues about the person's point of view: the object at hand, i.e. the most salient region in the person's field of view (referred to here as the "handled object"), and the place the person is located in (e.g., "office desk", "pharmacy", "coffee/tea desk"'). Complementary to global localization of the person from static cameras, the recognized place/location serves as a cut to where a person attention is directed, while object recognition allows a more semantically detailed decomposition of an activity. In daily living activity monitoring both egocentric vision detectors passes from complementary to essential when a person turns their back to the cameras.

Object recognition

We employ several detectors of "active objects" (objects either manipulated or most salient in the FoV of the user), as we consider that the identification of these objects is a crucial step towards activity understanding, as the recognition of activity-related objects adds more robustness to event models, especially when activities of daily living are the focus of interest. We consider one concept detector per object category, and the processing pipeline (Figure 9) is shared among all object detectors except for the nonlinear classification stage that is learned specifically for each object category.

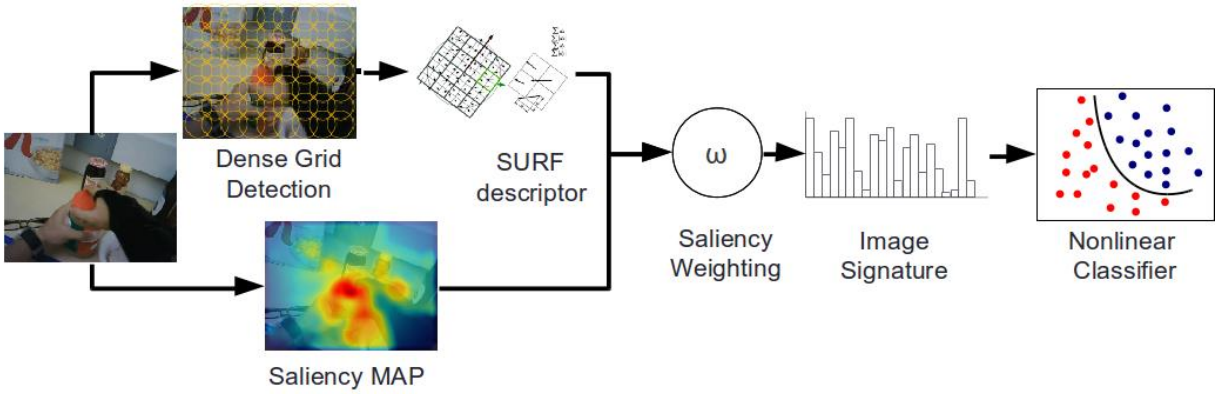


Figure 9. Processing pipeline for saliency-based object recognition in first-person camera videos

We have built our model on the well-known Bag-of-Words (BoW) paradigm [19] and proposed to add saliency masks as a way to provide spatial discrimination to the original Bag-of-Words approach. Hence, for each frame in a video sequence, we extract a set of *N* SURF descriptors d_n [7], using a dense grid of circular local patches. Next, each descriptor d_n is assigned to the most similar word *j* = 1..V in a visual vocabulary by following a vector-quantization process. The visual vocabulary is computed using a k-means algorithm over a large set of descriptors in the training dataset. We set the dictionary size to V=4000 visual words.

In parallel, our system generates a geometric spatio-temporal saliency map *S* of the frame with the same dimensions of the image and values in the range [0,1] (higher values denote greater salience). Details about the generation of saliency maps can be found in [11].

We use this saliency map to weight the influence of each descriptor in the final image signature, so that each bin *j* of the BoW histogram *H* is computed following the next equation:

$$H_j = \sum_{n=1}^{N} \alpha_n w_{n,j}$$

where the term $w_{n,j} = 1$ if the descriptor or region *n* is quantized to the visual word *j* in the vocabulary and the weight α_n is defined as the maximum saliency value S found in the circular local region of the dense grid.

Finally, the histogram *H* is L1-normalized to produce the final image signature. A SVM classifier [17] with a non-linear X^2 kernel [64] is then used to recognize the objects of interest over the weighted histogram of visual words. Using Platt approximation [54], we produce posterior probabilistic estimates O_k^t of the occurrence of an object *k* in the frame *t*.

Place recognition

Similarly to the recognition of handled objects, a closed-view recognition of the place the person is located in brings important information to the overall activity recognition process.

The general framework can be decomposed in three steps. First of all, for each image, a global image descriptor is extracted. We choose the Composed Receptive Field Histogram (CRFH) [57], since it was proven to produce good performance for indoor localization estimation [22]. Then a non-linear dimensionality reduction method is employed. In our case, we use a Kernel Principal Component Analysis (KPCA) [62]. The purpose of this step is twofold: it reduces

the size of the image descriptor, which alleviates the computational burden of the rest of the framework, and it provides descriptors on which linear operations can be performed. Finally, based on these features, a linear Support Vector Machine (SVM) [17] is applied to perform place recognition, and the result is regularized using temporal accumulation [22].

For the application considered in this chapter, each video is taken in a different environment. Consequently, our module has to learn generic concepts instead of specific ones, as is usually the case [22]. In this context, we need to define concepts relevant for action recognition, but as constrained as possible to achieve better performance. Indeed, for example the concept "stove" probably has less variability and may be more meaningful for action recognition than the concept "kitchen". Again, following the Platt approximation [54], the output of this module then is a vector P_j^t , with the probability of a frame *t* representing the place *j*.

4.2.3 People detection and tracking from color-depth sensor

People detection and tracking methods focus on detecting people present in scene images, and keeping track of their trajectories across time by appearance matching. To perform such tasks we adopted a standard vision pipeline, composed of people detection, tracking, and event recognition, but using a color-depth sensor instead of the standard static cameras. By incorporating depth information in the pipeline, the system becomes less sensitive to illumination changes, and can make use of more accurate real measurements of the scene, since real 3D information is available. As this module can localize people in space and time domain, it allows the multimedia framework to handle multiple people in the scene and derive person-centered events. Next subsections provide more details on people detection and tracking algorithms.

People Detection

People detection is performed through a depth-based framework algorithm described in [49]. This choice is due to the fact the algorithms offered in the libraries of Microsoft and of PrimeSense (the creator of Kinect depth sensor) cannot detect people at a large distance (over 5m) from the depth sensor due to the high variance of depth measurements. The adopted framework performs as follows: first, background subtraction is employed on the depth image provided by the color-depth camera to identify foreground regions containing both moving objects and potential noise. These foreground pixels are then clustered into objects based on their depth values and neighborhood information. Among these objects, people are detected using a head and shoulder detector. Tracking information is then used to find the correspondence between detected people in the current and previous frame. After this step, noise is removed based on the results of people classification and tracking. Finally, the background subtraction algorithm updates the background model according to the classification and tracking feedback. Details of the process can be found in [49].

People Tracking

The tracking algorithm takes as input the video stream and the list of objects detected in the current and previous frames in a sliding time window fashion. First, a link score is computed between any two detected objects appearing in this time window using a weighted combination of six object descriptors: 2D, 3D positions, 2D object area, 2D object shape ratio, color histogram and dominant color. Successive links form several paths, which an object can follow, within this temporal window. Each possible path of an object is associated with a score given by all the scores of the links it contains. The object trajectory is finally determined by maximizing the path score using a Hungarian algorithm.

In this tracker, the determination of the descriptor weights is a hard task because they depend on the content of processed video. Therefore we use a control algorithm [13] to tune these weights in an online manner. First, the tracking context of a video sequence is defined as a set of six features: the density of mobile objects, their occlusion level, their contrast with regards to the surrounding background, their contrast variance, their 2D area and their 2D area variance. Each contextual feature is represented by a codebook model. This model is learned in an offline phase where training video sequences are classified by clustering their contextual features. Each context cluster is then associated with best tracking parameter values using ground-truth tracking data. On running time (online control phase), once a context variation is detected, the descriptor weights are tuned using the learned values.

4.3 Multimedia Event Recognition

The multimedia event recognition (Figure 10) is responsible for linking world observations in the form of concepts into the ontology language representation, and then performing event inference from the observed conceptual data. Event inference is performed via a bottom-up approach where, once new knowledge is generated, it seeds the recognition of higher-level events. The main challenges of this module are to handle different data acquisition frequencies among sensors, their coarse time synchronization, the different levels of reliability across concepts and detectors, and to perform accurate event fusion under uncertain conditions (noise observations, incomplete evidence).

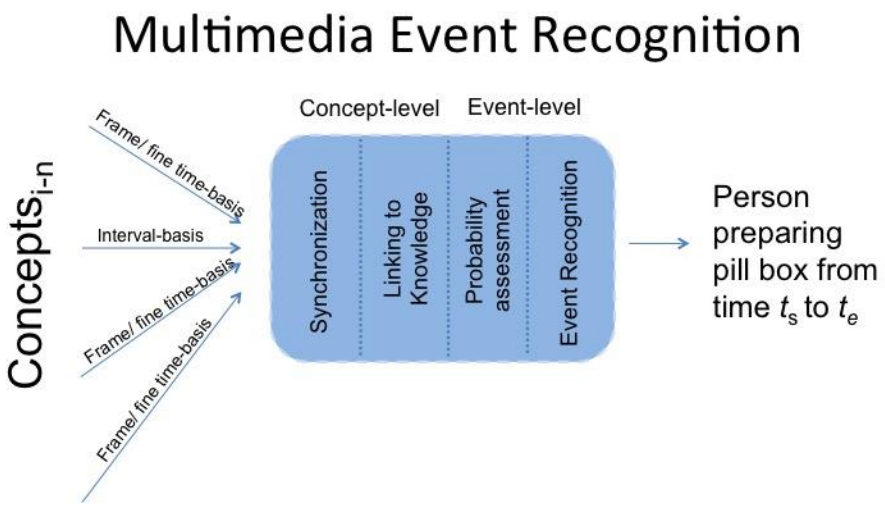


Figure 10. Multimedia Event Recognition workflow.

The abovementioned concept detectors for people, actions, and egocentric vision-derived concepts constitute the foundations of the multimedia framework, as they bridge the gap between the conceptual representation of event models provided by the knowledge-driven event representation and low-level sensor data. All detector outputs are integrated at a fine-grained representation (frame-wise resolution), to allow the framework to determine if an instantaneous or short-length recognition of a concept is caused by noise, or if the corresponding event time interval is naturally brief. Interval-based events (*e.g.*, from action detection) are processed as if they were frame-based, although their observations do not change in their event interval.

For event representation and recognition we extend the constraint-based framework proposed in [67]. The original framework is composed of an algorithm for temporal event recognition and a constraint-based language for event modeling in vision-based surveillance systems.

We model events follows an extension of their declarative and intuitive ontology-based language that uses natural terminology to allow end users (*e.g.*, medical experts) to easily add and change event models. Models are built taking into account *prior* knowledge of the experimental scene, and real world objects (herein called Physical Objects, *e.g.*, a person, a car, *etc*.) dynamically detected by underlying components of the vision pipeline.

Event models are composed of three parts:

- **Physical Objects** refer to real-world objects involved in the detection of the modeled event. Examples of physical object types are: mobile objects (*e.g.*, person, or vehicle in another application), contextual objects (equipment) and contextual zones (chair zone).
- **Components** refer to sub-events of which the model is composed.
- **Constraints** are conditions that the physical objects and/or the components should hold. These constraints could be logical, spatial and temporal.

We extend the set of physical objects of the constrained-based ontology language from the two basic physical object types: Mobile, and Contextual Objects, to five specializations necessary to describe the multimedia context: Person, Contextual Zones, Contextual Equipment, Handled Object and Place. Mobile is a generic class that contains the basic set of attributes for any moving object detected in the scene (e.g.,} 3D position, width, height). Person is an extension of Mobile class whose attributes are body posture and appearance signature(s). Contextual objects are sub-divided into contextual zone and equipment, handled objects, and places; and might consist of information from underlying components (*e.g.*, handled objects and places from wearable camera-related detectors), or a priori} knowledge about the scene, such as contextual zones and equipment (*e.g.*, furniture).

Constraints define conditions that physical object properties and/or components must satisfy, and they are classified into non-temporal, e.g., property-wise values, spatial and appearance constraints; and temporal constraints, such as the time ordering between two event components (sub-events). Temporal relationships are defined using Allen's interval algebra, *e.g.*, BEFORE, MEET, AND [1]. An alert clause can be optionally defined for each event model to rank it according to its importance for a specific task, e.g., visualization or triggering of an external process.

The ontology language hierarchically categorizes models according to their complexity as (in ascending order):

- **Primitive State** models a value of property of a physical object constant in a time interval.
- **Composite State** refers to a composition of two or more primitive states.
- **Primitive Event** models a change in value of a physical object's property (*e.g.*, posture), and
- **Composite Event** defines a temporal relationship between two sub-events (components).

Figure 11 presents the creation of the low-level event model *Person_inside_ZonePharmacy*, which models when a person's position lies inside the semantic zone "zPharmacy". This event

can be combined with other feature-based events (*e.g.*, posture) to model a higher-level event, like a person organizing their pill box.

```
PrimitiveState(Person_inside_ZonePharmacy,
 PhysicalObjects( (p1: Person),(zPharm: Zone))
 Constraints(
   (p1->position in zPharm->Verticies)
 Alarm ((Level : NOTURGENT)) )
```

Figure 11. Composite Event Model for "Prepare pill box" based on a single source of concepts

4.3.1 Concept and Event Modeling

We have modeled the set of outputs of each detector as physical objects of the constraint-based ontology language described above. For detectors where the conceptual information does not map to a physical object, we link their observations as instances of primitive states (lowest-level events). The linking between concepts and the ontology language is performed as follows:

- 2D camera action detection is linked as primitive state instances, since its detector cannot identify the author of the motion.
- Wearable-sensor derived concepts are linked as handled objects and places.
- Contextual zones correspond to the decomposition of the 3D projection of the scene floor plan into a set of 3D spatial polygons that carry semantic information about the monitored scene (e.g., zones like "V", "armchair", "desk", "coffee machine"). Zones are manually defined on the 3D coordinate system of the color-depth sensor.
- Concepts from the people detection and tracking detector from color-depth sensor are linked as instances of the physical object "person".

The defined contextual zones are telephone desk, drink preparation desk, reading low-table, pharmacy, and plant. The contextual zones and people physical objects obtained from the color-depth camera are combined early in the inference step to derive low-level events (primitive states) about the person's localization in the scene and the posture of the person, and subsequently composite events from these events, as early estimations of daily living activities. From here on we referred to these knowledge-driven early estimations of ADL as observations of the complex event recognition detector.

Wearable handled object recognition (OR) detectors generate estimations over 18 visual concepts, each one ranging in the probability interval [0-1] with different performance recognition rates: accounts, basket, bills, checks, activity instructions, kettle, map, medical instructions, pen, telephone, pill box, plastic glass, remote, TV, tablet, teabag, tea box, and watering can. Place detector follows the same idea and provides probability estimations of place in the field of the view of the wearable camera. It focuses on places of the following event classes: medication, telephone, desk, and tea. Place physical objects differ from contextual zones in the sense that the latter is related to the global position of the person in a third person view of the scene, while the first corresponds to the focus of the person's field of view. This distinction proves useful to distinguish a general-purpose activity (e.g., reading

instructions) from expected activity from a person located on a defined semantic zone of the scene (e.g., coffee table).

Action recognition detector (AD) provides estimations about a set of mutually exclusive atomic actions: answer phone, call phone, look on map, pay bill, prepare drugs, prepare drink, read paper, water plant, and watch TV. The identification of who is handling a given object or performing a certain action in the scene is iteratively solved as the concepts without person identification are combined with event model instances based on data from people detection and tracking detector, by what we can call, semantic closeness. For instance, the person holding a kettle is most likely to be the person currently at the prepare drink area than the one in the living room.

Figure 12 presents an example of high-level composite event, the model *PreparePillBox_HL*, that describes - using the ontology language - how to recognize a person preparing their pill box. The model has four physical objects: a person, a 3D semantic zone, a handled object, and a place. It is composed of two sub-events PreparePillBox_CER, a high-level event modeled from the features of people detection and tracking (*e.g.*}, Figure 11); and the event *PreparePillBox_AD* from action detection module.

Figure 13 illustrates an overview of the bottom up inference process using the ontological representation of event hierarchy, and the iterative process of linking low level concept instances into intermediate events, and subsequently into high-level events, like *PreparePillBox_HL*.

```
CompositeEvent(PreparePillBox_HL,
PhysicalObjects( (p1: Person),(zPharm: Zone), (PillBox: HandledObject),
(medication_pl: place))
 Components(
   (c1: PrimitiveState PreparePillBox_AD() )
   (c2: CompositeEvent PreparePillBox_CER(
                              p1,zPharm)))
 Constraints(
   (c1->Interval AND c2->Interval)
   (duration(c2) > 3))
 Alarm ((Level : URGENT))
)
```

Figure 12. High-level Composite Event Model for “Prepare pill box” based on concepts from different visual modalities

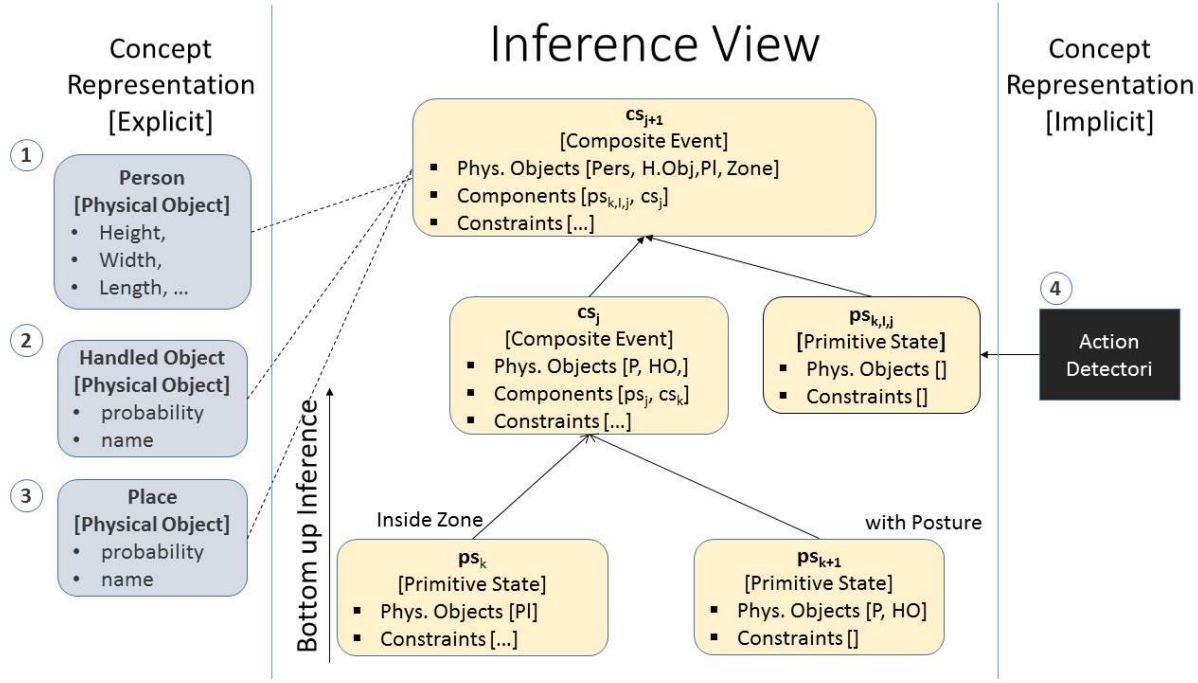


Figure 13. Diagram of the linking of the sensor-derived concepts into ontology language physical objects and low-level events.

4.3.2 Concept Stream Synchronization

Data fusion of heterogeneous sensors generally assumes coarsely time-synchronized sensors and operates on a time window manner that might be static or dynamically determined [38]. In such approaches the size of the time window is an important parameter that needs to be optimized to cover the maximal time an event presents and also provide some flexibility to handle small temporal shifts between sensor event streams. These approaches tend to introduce uncertainty on the boundaries of event temporal intervals, generally in the form of overestimation. This situation is more appropriate when targeted events have similar duration, since heterogeneous duration would cause instantaneous and short-lengthened events to be overlooked. To overcome such limitations, we perform sensor data synchronization but at conceptual (semantic) level, to overcome concept stream misalignment and allow for a more fine-grained event modeling, we propose a novel approach based on stream alignment based on semantic (concept) similarity.

Given two sensor concept streams, the concept stream to be aligned (A) and the reference stream (B), we perform semantic temporal alignment between these streams as follows:

1. The concepts in the sensor streams A and B are encoded with a unique label representing the composite event they are part of (*e.g.*), given the composite event "Water Plant" has label 8, all its related concepts, like "plant" and "watering can", will be encoded with the same label.
2. Dynamic time warping (DTW) [60][47] is then performed to find the optimal alignment (warped streams) of A and B,
3. Then, we compare B to its warped version to identify the added time positions and prune them from the warped A, projecting A stream back onto the original temporal axis of B.

4. Finally, we perform median filtering on the back-projected version of B to remove spurious concepts introduced by the procedure.

The outcome of the described procedure is a new stream, optimally aligned with respect to common concepts, but also carrying sensor exclusive concepts that were not temporally coherent before. Concepts are projected onto the time axis of the people detection and tracking modules, since this module is the only one capable of retaining the identity of the monitored people over time, while also utilizing an intermediate temporal resolution among sensors.

For probabilistic concept detectors based on the same modality (*e.g.*, wearable camera) where we do not have binary detection of a concept but estimations on a probability interval, we adopt the following procedure: we first generate a main concept stream based on the most likely concept at each frame among all concept-related streams (*e.g.*, handled objects), and then proceed with general steps 1-4 over the generated stream. Once the general steps are completed, we utilize the time series deformation performed into the main stream to align the individual concept detector streams.

The described methodology makes the following assumptions: each concept cannot be part (contribute positively) of more than one (composite) event; several concepts may contribute to a single composite event; sensors are, at least, coarsely time synchronized.

4.3.3 Concept and Event Probability Estimation

In multimodal video content analysis and retrieval it is common to build a set of classifiers - basic or not, from different representations of video data - for each event class, and then average their output to obtain the final score for an event recognition [47]. Such approaches target the output overlap between redundant detectors of an event, and, at least implicitly, hypothesize all sensors/detectors have equal performance at the recognition of the event class of interest.

These assumptions do not hold for the targeted setting, since concept detectors are learned from different sensors that have different perspectives of the monitored scene. These differences in perspective directly affect the performance of such detectors, both positively and negatively. Consequently, adopting a fusion strategy that weights the contribution of different sensors for an event in an equal basis will most probably not provide the optimal solution.

To cope with differences in sensor performance at concept detection, different contribution of each concept for a high-level event, and the uncertainty of a concept detection at each frame, the multimedia event recognition framework models each concept taking into account three probability variables: reliability given sensor, relevance given composite event, and recognition probability.

Concept recognition probability is dynamically estimated and corresponds to the confidence a concept detection has in the recognition of the targeted concept. Reliability corresponds to the learned accuracy of detector at generating concept instances during the realization of the targeted high-level event. This variable corresponds to the number of time units the concept is recognized by the respective sensor with respect to the number of time units the high-level event they refer to is observed from the color-depth sensor. Relevance is the conditional probability of the composite event given the concept.

The reliability and relevance values are obtained by a supervised learning step using Equations 1 and (2, respectively, based on ground-truth data for composite events. Reliability corresponds to the detector performance (*e.g.*, F1-score, precision) at recognizing the given concept during the realization of a composite event. For instance, Equation 1 denotes reliability in terms of precision index of performance.

$$RLB\left(ce_{k,i,j}|cs_j\right) = \frac{|TP|}{|TP| + |FP|} \quad (1)$$

where:

- $RLB\left(ce_{k,i,j}|cs_j\right)$, reliability of concept *k* from detector *i* given its composite event cs_j,
- $|TP|$, number of times concept ce_k is correctly recognized by concept detector *i* during an instance of event cs_j in event ground-truth data (events annotated by a domain expert),
- $|FP|$, number of times $ce_{k,i,j}$ is observed by the concept detector.

The relevance of a concept for a given composite event Equation (2) corresponds to the conditional probability of the composite event given the concept.

$$\mathrm{RLV}(cs_j|ce_{k,i,j}) = \frac{|\, ce_{k,i,j} and\ cs_j\,|}{|ce_{k,i,j}|} \quad (2)$$

where:

- $\mathrm{RLV}(cs_j|ce_{k,i,j})$, frequency that concept $ce_{k,i,j}$ is detected during an instance of composite eventcs_j in event ground-truth data,
- $|ce_{k,i,j}$ and $cs_j|$, number of times $ce_{k,i,j}$ is present (recognized) during an instance of cs_j in ground-truth data,
- $|ce_{k,i,j}|$, number of times $ce_{k,i,j}$ is observed.

The relevance and reliability of each concept are computed based on the overlap between instance of concept detector with composite events annotated by a domain expert (ground-truth data) on the time axis of the reference sensor (here, color-depth sensor).

The probability of a composite event is modeled as function of its parts (sub-events and physical objects) to target the composite nature of events. To combine the probabilities of the parts we adopt a countable mixture model (Equation (3). Briefly, each concept (low-level event or physical object) is considered as a random variable ($ce_{k,i,j}$), and we employ a Countable Mixture Distribution (CMD) to combined these part probabilities into the composite event model. The weight of each concept corresponds to its combined reliability and relevance. The probability of a concept is the probability of the current observation in case of a probabilistic detector, and a binary indicator function otherwise (0,1). Equation (4 denotes the probability of a composite event cs_j. Equation (5 denotes the partition function that normalizes the reliability and relevance of an event.

The concepts that are involved in an event class are obtained from the event models defined by domain experts using the ontology language.

$$f(x) = \sum_{i=1}^{N} w_i * P(x_i)$$
$$w_i >= 0 \qquad (3)$$
$$\sum w_i = 1$$

$$\frac{1}{Z}\sum_{i=1}^{N}(\text{RLV}(cs_j|ce_{k,i,j}) + RLB\left(ce_{k,i,j}|cs_j\right)) * P(ce_{k,i,j}) \qquad (4)$$

$$Z = \sum_{ce_{k,i,j} \in cs_j} \text{RLV}(cs_j|ce_{k,i,j}) + RLB\left(ce_{k,i,j}|cs_j\right) \qquad (5)$$

where:

- $\text{RLV}(cs_j|ce_{k,i,j})$: conditional probability of composite event j given concept k from detector i,
- $RLB\left(ce_{k,i,j}|cs_j\right)$, reliability of concept $ce_{k,j}$ given detector i,
- $P(ce_{k,i,j})$: probability of concept ce_k from detector i and part of composite event j.

The outcome of this step is the confidence of the multimedia framework on each event class given the observations of the real-world provided by the visual concept event detectors. This confidence is computed based on the composite nature of high-level event and its parts (physical objects and lower-level events).

4.3.4 Semi-Probabilistic Event recognition

This step is performed based on an extension of the deterministic temporal algorithm proposed in [67], initially proposed for single camera systems. The present approach takes as input concepts from the visual concept detectors of the multimedia framework and then links them as physical objects and primitive states of intermediate- to high-level event models of the knowledge-driven approach. To such concepts, it adds a probability (or confidence) value computed using the procedure defined in subsection 4.3.3. From these quantified observations of the real world we infer new knowledge (events) based on a priori defined ADL models and sub-models, following an iterative and hierarchical approach on a bottom-up fashion.

Event instance probabilities are taken into account to handle event ambiguity over mutually exclusive high-level events and to suppress low-probability events. Maximum a posteriori is used as described in Equation 6 to decide upon a set of mutually exclusive candidates for events. For instance, IADLs are mutually exclusive and only the most probable one must be kept. A probability threshold th_{cs_j} is used over event instances to ensure that low probability events are not recognized. This threshold approach is determined per event class, and also allows the system to infer when none activity is in progress.

$$cs = \begin{cases} argmax_{cs_j} P(CS), if\ P(cs_j) > th_{cs_j} \\ \emptyset, otherwise \end{cases} \qquad (6)$$

where,

- cs: most likely composite event,

- CS: set of mutually exclusive composite events in analysis,
- th_{cs_j}: minimum probability for the recognition of composite event cs_j.

The inference task is only performed over the semantically aligned sensor concept stream. For handled objects and places the framework takes advantage of their individual probability estimations to choose only the most probable concept at each time *t*. For actions detection and PDT-derived events that do not carry such estimations, we employ an indicator function (1 for detection, 0 otherwise) weighted by the concept reliability and relevance to the targeted event model.

In summary, the inference process takes into account the instances of the visual concept detectors on a frame-basis, and fuses them using the knowledge-driven event model templates. These templates allow the inference step to focus on the most important concepts for each targeted activity (in terms of physical objects and sub-events), reducing its complexity only to the subset of most probable events. Probability estimations of composite events enable the framework to handle ambiguous situation and to quantify the likelihood of each activity instances. Logic and temporal constraints can be used throughout the event inference step to impose real world constraints into event models.

4.4 Experiments

The evaluation of the proposed multimedia event recognition framework is performed as follows: firstly, we evaluate the effects of the concept synchronization approach over the performance of the concept detectors, secondly, we focus on the fusion problem and compare the results of the proposed approach to the individual detectors, and to two baselines methods: Ontology-based Semantic Interpretation (OSI, subsection 4.4.2), and a fusion method based on Support Vector Machine algorithm -- a standard algorithm for classification (Subsection 4.4.3).

All evaluations are performed on multimedia recordings of older people naturally undertaking daily living activities during a clinical protocol for a medical study of dementia (subsection 4.4.1). All experiments are performed on an offline fashion, but only OSI considers the complete set of concepts recognized during the multimedia recording in its reasoning process. Results are presented on the validation and test sets of a 10-fold cross-validation scheme used to learn the model parameters for the proposed approach and the SVM baseline.

4.4.1 Data set: monitoring IADLs of older people

Participants aged 65 years and above were recruited by the Memory Center (MC) of Nice Hospital. Inclusion criteria of the Alzheimer Disease (AD) group are: diagnosis of AD according to NINCDS-ADRDA criteria and a Mini-Mental State Exam (MMSE) score above 15. AD participants who have significant motor disturbances (per the Unified Parkinson's Disease Rating Scale) are excluded. Control participants are healthy in the sense of behavioral and cognitive disturbances. The clinical protocol asks the participants to undertake a set of physical tasks and Instrumental Activities of Daily Living in a Hospital observation room furnished with home appliances [26]. Experimental recordings used a color-depth camera (Kinect®, Microsoft©, ~ 10 frames per second), a GoPRO Hero - first generation - for wearable camera, and a 2D-RGB static camera (AXIS®, Model P1346, 8 frames per second).

The activities of the clinical protocol are divided into two scenarios: Guided, and Semi-guided. Guided activities (10 minutes) intend to assess kinematic parameters of the participant's gait profile (e.g., static and dynamic balance test, walking test). Semi-guided activities (15 minutes) aim to evaluate the level of autonomy of the participant by organizing and carrying out a list of instrumental activities of daily living (IADL). The participant enters the observation room alone with the list of activities to undertake, and he/she is advised to leave the room only when all required activities are completed.

The clinical protocol IADLs are following:

- watch TV,
- prepare tea/coffee,
- using the telephone (calling, answering),
- read the newspaper/magazine,
- water the plant,
- prepare medication (organize pill box),
- manage finances (write a check, establish account balance),

The evaluation of the multimedia approach focuses on the recognition of activities of daily living of the semi-guided scenario. Figure 14 illustrates the observation room where the participants undertake IADLs and the semantic zones that are annotated to incorporate a priori knowledge of the scene into the knowledge-driven approach.

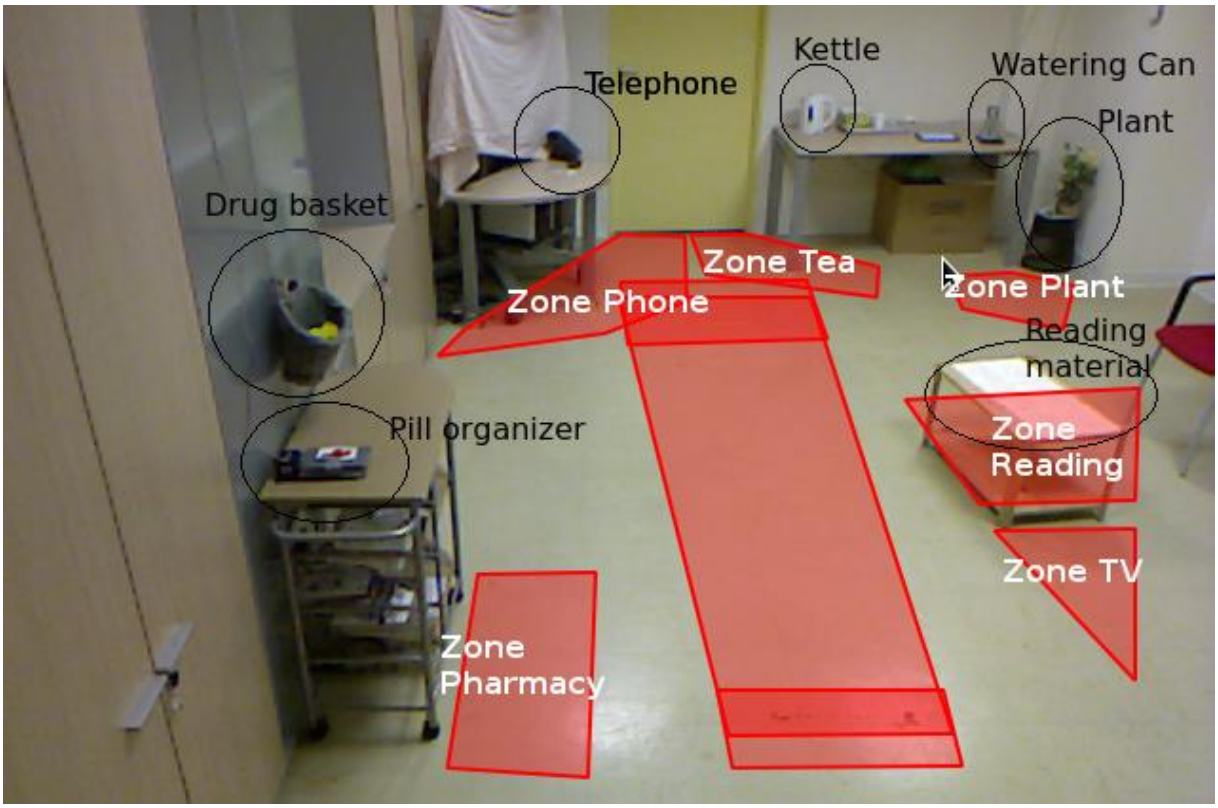


Figure 14. Observation room where daily living activities are undertaken.

4.4.2 Baseline 1: Ontology-based Semantic Interpretation

The ontology-based framework for multi-sensor fusion [44] is based on the use of RDF/OWL [31] ontologies to capture the dependencies among low-level domain observations and

complex activities (events). More specifically, following a knowledge-driven approach, it defines the Context Dependency Models of the domain that capture the background knowledge required to detect the complex activities. The context dependency models serve as input to the semantic interpretation procedure for the recognition and classification of complex activities. The objective of the interpretation procedure is to analyze traces of observations provided by the various modules of the application domain and group them into meaningful situations, classifying them as complex activities. The interpretation algorithm consists of three steps: (a) definition of partial context, (b) identification of contextual links and (c) recognition and classification of situations. Details about OSI approach are available in [44].

The ontology-based semantic interpretation serves as a baseline for the delimitation of the temporal boundaries and the recognition of events if a holistic view of the concepts of a multimedia recording is employed. Its limitations are the following: it cannot handle interleaved activities, nor can it resolve conflicts after the recognition process. It also does not handle dynamic and incremental generation of partial contexts and context links in a (near) real-time activity recognition, as it operates with a global view of all recognized events for a given multimedia recording. Finally, this baseline approach does not handle uncertainty in the input data, and assumes all observations (primitive and high-level) have the same confidence (100 %).

4.4.3 Baseline 2: Support Vector Machine

The second baseline consists of a set of linear SVM classifiers, tasked to learn and recognize activities from the combination of observations from all available sensor concept streams using a time-window. The rationale behind this baseline is to assess the any differences in event recognition performance between a fully supervised approach working over an intermediate representation of raw sensor data (CER events, OR objects and places, and AD detector actions), and the proposed approach that considers different aspects of each concept during its fusion into high-level event models.

The input for these classifiers is the normalized histogram of composite event concepts across sensors. Similar to the proposed concept synchronization method, we analyze each sensor concept stream, and encode its concepts with the label of the composite event they are part of. Then we compute a histogram of the composite event concepts across all encoded sensor concept streams within a time window. The outcome of this process is an intermediate representation about the amount of evidence available for the recognition of each composite event. In the training set, histograms are computed from the exact event intervals provided by ground-truth data. For validation and testing sets we employ a continuous temporal sliding window that spans from the frame in analysis t back to t-w frames in the past. The size of the time-window is determined by experimentation, and it is tested with the average lengths of the activity classes in the data set. One-versus-all scheme is employed to learn the classifiers.

SVM parameter and time-window size are evaluated on the training and validation sets of same 10-fold cross-validation scheme used by the proposed approach to learn its parameters, and are chosen based on the respective SVM performance in the validation set. SVM results are computed over the concept streams aligned with the proposed synchronization method.

4.4.4 Evaluation

Two main evaluations are performed to demonstrate the framework contributions at synchronizing heterogeneous sensors, and at event recognition by fusing multimodal concepts streams based on their reliability and relevance. For each evaluation, we perform a frame-wise comparison between ground-truth data and the proposed approaches. We count as true positive the number of frames a concept detector output agrees with the ground-truth data, false positive when the concept is detected but there is no event in the ground-truth, a false negative when there is an event in ground-truth data, but none for the concept/event detector, and finally, both a false negative and a false positive when the concept detector and ground-truth disagree. This strict comparison between visual concept data/events and ground-truth data is necessary since we aim at determining the contribution and reliability of each individual detector at the highest resolution possible, in this case at frame level. By handling events and concepts at this level, instead of at interval representation, we may observe their local probability and its variations, and then determine their reliability more accurately.

For the evaluation of the semantic-based concept synchronization technique, we employ the action detector (AD), the complex (or composite) event recognition detector (CER), and the object recognition (OR) at the recognition of composite activities, as a way to quantify the gain in performance of their aligned version compared to their original form. The performance of their warped version with the ground-truth concept stream is also provided to evaluate the step that back-projects the warped sensor concept stream onto the axis of the targeted stream.

For the evaluation of the overall framework for multimedia event recognition we present results using validation and test sets, with F1-score as performance measure, comparing the framework performance with and without probability thresholding to individual detectors, with and without alignment, and to baseline approaches.

4.5 Results

4.5.1 Concept Stream synchronization

Figure 15 illustrates an example of alignment between a concept stream generated from the events annotated by domain experts (GT) - using the color-depth sensor images as reference - and a concept stream of the action detection (AD) module using the proposed technique. We may observe that the proposed technique accurately translates the AD stream from its original form - coarsely synchronized - to a new form that is optimally synchronized with the reference stream and also preserves most of the shape characteristics of the original AD concept stream.

Table 3 presents a quantitative evaluation of the gain in performance obtained by aligning the concept detector streams. To measure this improvement we measure the ability of each concept detector (action, AD; complex event recognition, CER; and object recognition, OR) at recognizing alone the composite event they are part of. We present results on three cases: the original concept streams; the warped case, where both ground-truth and sensor stream are optimally aligned, and at last, the detector performance after being aligned by the semantic-based synchronization, compared to the original ground-truth data. Performance is measured using F1-score index.

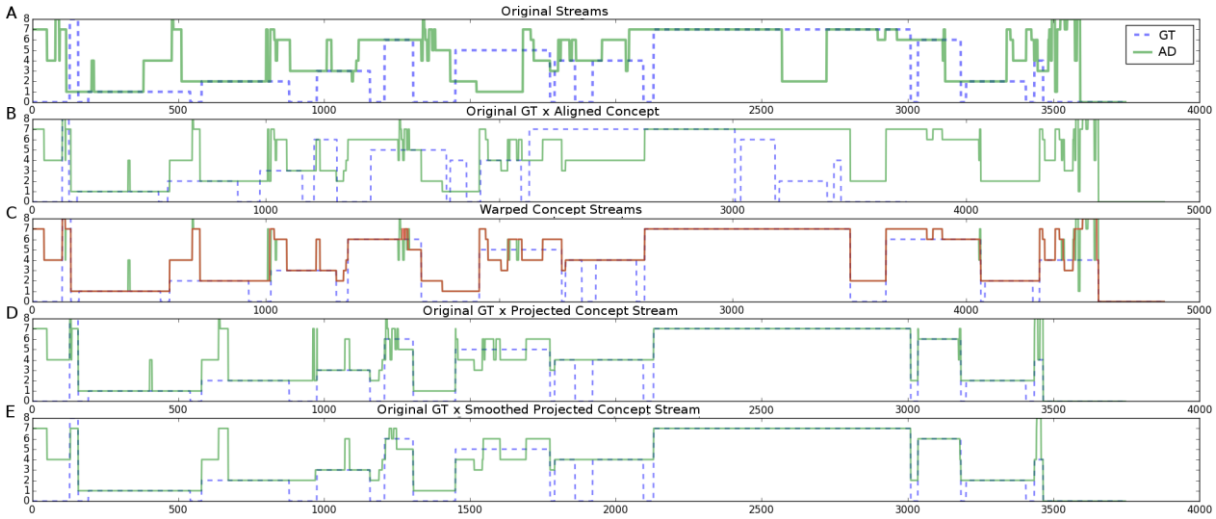


Figure 15 Semantic alignment between the concept stream of the action detector (AD) and a concept stream (GT.

Table 3 Composite Event Recognition from concept detectors (F1-score, %)

	Detector / Stream alignment								
	AD			**CER**			**OR**		
IADLs	NA	WS	WBS	NA	WS	WBS	NA	WS	WBS
S. Bus line	40.9	44.3	45	16.6	27.7	27.8	11.7	13.7	13.7
M.Finances	61.7	60.9	62.1	0	0	0	26.7	30.9	28.7
P. Pill box	49.4	55.3	57.1	69	61.8	62.6	23.8	24.5	21.7
P. Drink	31.6	51.4	49.2	71.9	86.6	85.9	0	0	0
Read	50.8	62.9	56.8	73.8	97.9	98.2	0.1	8.3	7
T.Telephone	38.9	66.5	60.9	68.2	83.9	83.3	13.7	14.2	13
W. TV	17.2	42.9	36.5	9.97	30.5	27.3	10.1	17.1	14.7
W. Plant	9.04	21.4	21.9	47.4	86.4	86.4	0	0	0

N: 17 participants; 15min.each; Total: 255min. (-) denotes concepts not available for the detector. AD: action detector, CER: Composite Event Recognition, OR: Object recognition. NA: Not aligned, WS: warped and smoothed, WBS: warped, backprojected and smoothed.

The proposed synchronization method improves the performance of the CER detector when compared to its original stream, with exception of the "prepare pill box" event. It also has a higher performance than its warped version in three out of seven classes, and a very close performance otherwise (e.g., "prepare drink", "talk on the telephone", and "watch TV" events). In the action detector case, the synchronized streams perform better than the original stream for all cases, but the warped streams have better performance than the synchronized version in half of the classes ("prepare drink", "reading", "talk on the telephone" and "watch TV" events). Finally, for OR detector the aligned stream again outperforms the original stream form for all cases, except for two event classes: "prepare pill box" and "talking on the telephone". In this detector case the warped streams still outperforms the synchronization method for the majority of cases.

4.5.2 Concept Fusion for Event Recognition

Figure 16 presents the performance of the proposed multimedia event recognition framework according to the probability threshold adopted to filter spurious event instances. Performance is presented as the mean F1-score among all event classes and folds of the validation set. We may observe most event classes present their highest recognition rates adopting probability threshold values between 0.4 and 0.5. Exceptions are "search bus line" and "talk on telephone" events where the threshold value of 0.1 provides the highest performances.

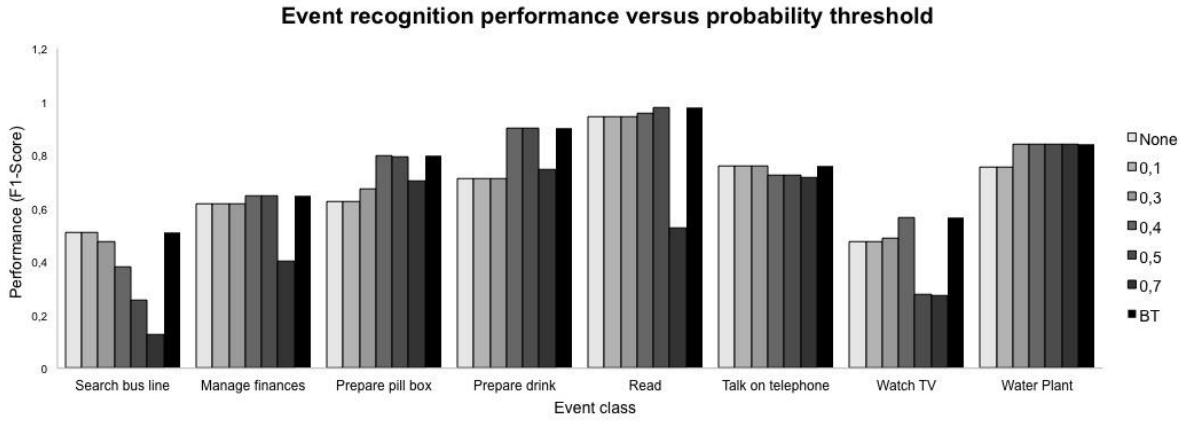


Figure 16. Event recognition performance according to probability threshold.

Table 2 presents the performance of the multimedia event recognition framework (with and without probability thresholding) and of its individual concept detectors in the validation set, before and after semantic synchronization.

Table 4 Event recognition performance in the validation set

mean F1-score	Stream Alignment / Detector							
	None			Aligned			Proposed	
IADLs	CER	AD	OR	CER	AD	OR	WT	BT
S. Bus line	13.1	47.3	8.3	13.9	45.3	17.8	51.1	51.1
M.Finances	0	73	24	0	66.7	27	61.8	64.7
P. Pill box	71.4	55.1	21.4	66.3	56.7	24	62.4	79.7
P. Drink	77.6	37.2	0	91.6	53.5	0	71.2	91
Read	73.2	49.9	0	98.2	54.8	0.5	94.5	97.7
T.Telephone	65.9	44	14.4	89	62.6	14.9	75.8	75.8
W. TV	13	22.4	11.9	30	44.9	17.6	47.6	56.6
W. Plant	45.3	11	0	83.4	26.7	0	75.6	84.2

WT: without probability thresholding,
BT: event recognition performance of the best thresholds

Results show the proposed framework has a performance higher than the aligned versions of its individual detectors at the recognition of IADLs, with two exceptions: "managing finances" and "talking on the telephone" events. For the first event, the stream without alignment of the action detector has a performance 9 % higher than the proposed method,

while for the second event the aligned version of CER detector has a performance 14 % higher. Probability thresholding improves the event recognition in the majority of cases if not it performs equal to its counterpart without it.

Table 5 compares the performance of the proposed framework to the individual concept detectors in the test set samples, before and after semantic synchronization. The proposed framework outperforms the individual sensors in all cases and classes, with exception of aligned CER in the events “talk on the telephone” (-17.5 %), "reading" (-2.8 %), and "search bus line" (-1 %).

Table 5 Event recognition performance in the test set

mean F1-score	**Stream Alignment / Detector**						
	None			**Aligned**			**Proposed**
IADL	CER	AD	OR	CER	AD	OR	BT
S. Bus line	28.6	19.6	23.2	74.1	43.8	0	73.1
M.Finances	0	27.4	37.6	0	43.7	35.4	43.7
P. Pill box	60.6	28.6	32.4	49.1	58.7	24.7	65.0
P. Drink	43.7	4	0	57.5	27.6	0	64.0
Read	77.2	56.2	0.6	98	68.9	45.9	95.2
T.Telephone	77.6	18.7	10.7	93.1	54.2	5.2	75.6
W. TV	0	0	4.3	18.5	8.4	5.1	35.8
W. Plant	56.8	0	0	100	0	0	100.0

Table 4 Comparison of the proposed approach to the two baselines methods.

Table 6 Comparison to baseline methods in the test set

	Fusion Approach		
	Baselines		**Ours**
IADLs	SVM	OSI	
Search bus line	44.2	21.6	51.1
Manage finances	44.0	0.0	64.7
Prepare pill box	45.9	71.3	79.7
Prepare drink	20.0	68.7	90.1
Read	90.2	70.3	97.7
Talk on telephone	72.1	2.7	75.8
Watch TV	2.3	0.0	56.6
Water Plant	0.0	16.1	84.2

OSI: Ontology-based Semantic Interpretation

4.6 Discussion

4.6.1 Concept Stream Synchronization

We have proposed a method for heterogeneous visual sensor alignment based on semantic similarity. Results at event recognition level show that detectors semantically aligned outperform their original form and warped versions in the majority of cases, with very few exceptions. This demonstrates the method's capability of translating accurately the optimal alignment achieved by the warped streams to the time axis of the reference stream (sensor).

For the cases when the synchronization method performs worse than the warped concept stream, this behavior is due to a loss of information during the projection from the warped stream space onto the temporal axis of the reference concept stream. This loss happens when DTW removes/adds time points in stream regions with high concept variation. Consequently, the suppression of time points in this region penalizes severely the performance of one event class in favor of another. This is specially the case for object recognition stream, where there are object estimations for every frame. This decay in performance also occurs for naturally brief concepts, which tend to be removed when temporally close to longer concepts shared among streams.

Finally, when the original concept stream performs better than the synchronized and the warped streams, results suggest the DTW algorithm could not achieve the optimal alignment between the two streams.

4.6.2 Concept Fusion for Event Recognition

We have presented a multimedia event recognition framework based on the semantic synchronization of different visual sensors, a knowledge-driven approach for multimodal concepts representation and event definition, and a semi-probabilistic method to estimate concept reliability and relevance to support a more accurate concept fusion and event recognition.

Results demonstrate the proposed framework outperforms all baseline approaches in the test set of the 10-fold cross-validation, by handling incomplete and noisy observations. OSI baseline presents a similar performance to the proposed approach on activities like “read”, “prepare pill box”, and “prepare drink”, and outperforms SVM baseline on the last two events. The reason behind the higher performance of OSI compared to SVM is the existence of conceptual data from all detectors for these events. The same performance is not observed for “manage finances” event, where concept information is only available for AD and OR detectors, since this event happens outside of the field of view of color-depth sensor. Similarly, a drop in performance is seen for “watch TV” event, since this event is undertaken at the border of the field of view of the color-depth sensor, which tends to generate noisy observations from CER detector, and consequently compromise OSI performance due to its limitation at handling uncertain information. The best results of SVM-baseline happens in “Read” and “Talk on the telephone” events. Although this baseline had lower performance than the proposed framework, it performs better than OSI for “Search bus line”, “manage finances”, and “prepare pill box” events. The higher performance of this baseline for such cases highlights its capability to implicitly learn how to handle incomplete evidence, although not as accurately as the proposed approach.

Both OSI and SVM baselines underperform on the presence of naturally brief activities, like “water plant”. For the first baseline, this performance may be attributed to the noisy and low

reliability of AD detector observations for this event. For the SVM baseline, the low performance is mostly due to the usage of time window approach, which tends to attenuate the evidence of brief events in detriment of longer ones.

We also observed that the proposed framework outperform its individual detectors in the majority of events. This evidence shows that the framework is capable of performing sensor concept fusion in a meaningful way by modeling individual detectors' reliability at each concept and the concept relevance to its corresponding event model. For a very few cases, the performance of aligned individual detectors was higher than the fusion approach. The lower performance in such cases is mostly due to a dynamic component of sensor reliability that is not captured by the static representation of sensor performance. To solve these cases the reliability weights should be adapted dynamically according to a variable measure of sensor reliability that could allow the system to profit of the full potential of an individual detector performance, without compromising the overall system performance.

5 Situation Descriptors, Context Connections and SPARQL Rules for Knowledge-Driven Activity Recognition

Ontologies have attracted growing interest as means for modelling and reasoning over contextual information and human activities in particular. Under this paradigm, OWL has been used in a substantial body of work for describing the elements of interest (e.g. events, objects, location), their pertinent logical associations, as well as the background knowledge required to infer additional information. In many cases, activity recognition is further augmented with rules for representing richer relationships not supported by the standard ontology semantics, like e.g. temporal reasoning and structured (composite) activities.

In this direction, we proposed a context-aware activity recognition framework that introduces the notions of situation descriptors and context connections [45]. The key idea is to use ontologies for representing dependencies among high-level situations and low-level observations in a loosely-coupled manner, rather than defining strict contextual patterns that cannot provide enough flexibility for handling the imprecise and ambiguous nature of real-world events. The contextual information encapsulated inside the dependency models is used for identifying links among RDF observations that signify the presence of complex activities and to subsequently classify them as high-level activities.

In this section, we describe the SPARQL-based implementation of situation descriptors and context connections. More specifically, we describe the ontologies that can be used for modelling information in various levels of abstraction and we elaborate on the architecture and technologies that underpin the implementation.

5.1 Representation Layer

Domain information can be modelled in two levels of abstraction: events and situation descriptors.

5.1.1 Events

We use the term “event” to refer both to low-level observation types (e.g. location, objects) and complex activities (e.g. prepare tea). The Event Model hierarchy is depicted in Figure 17 and provides the lightweight vocabulary for capturing basic event-related information, such as, event hierarchies and temporal extension. More specifically, the Event Model extends the leo:Event concept of LODE [63] to benefit from existing vocabularies and data sources that describe events. em:Event is the root class with two direct subclasses em:Observation and em:Activity for modelling observations and activities, respectively. The em:Observation class is further extended with four additional observation types for modelling locations (e.g. in table zone), postures (e.g. sitting), actions (e.g. drinking) and objects (e.g. cup), which are the basic building blocks of observation types that are currently supported.by the implementation.

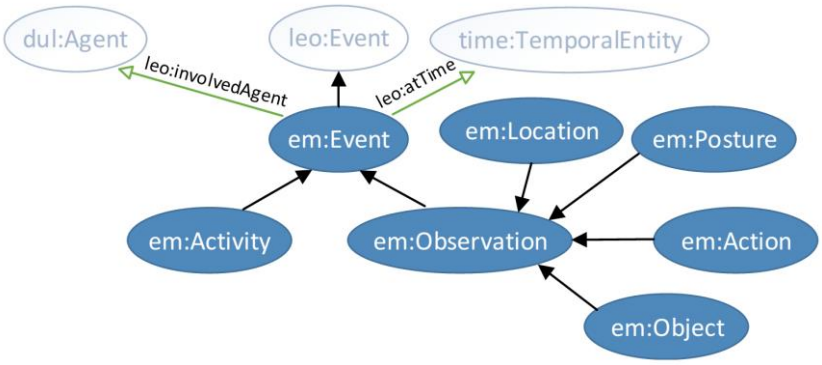


Figure 17. The upper level Event Model.

Information about the actors and temporal extension of events is captured using the LODE properties leo:involvedAgent and leo:atTime, respectively. For example, the observation about the detection of the cup object (em:Cup) is modelled in the Event Model as:

```
:cup1 a em:Cup ;
  leo:atTime :t1 ;
  leo:involvedAgent [a dul:Agent].
:t1 a time:TemporalEntity
  time:hasBeginning [
    a time:Instant ;
    time:inXSDDateTime "2015-01-02T18:06:46"
  ];
  time:hasEnd [
    a time:Instant ;
    time:inXSDDateTime "2015-01-02T18:16:12"
  ].
```

5.1.2 Situation Descriptors

The situation descriptors describe in a loosely-coupled manner the background knowledge required to detect the complex activities of the domain. They are defined in terms of the lightweighted ontology pattern depicted in Figure 18 that consists of the following knowledge structures:

- `SituationDescriptor`: Top-level container class for storing contextual correlations.
- `dependency`: Property that designates the low-level observation types of the dependency.
- `describes`: Property that designates the complex activity of the descriptor.

The sd:SituationDescriptor class extends the dul:Situation class of DUL and operates at the meta-model layer, i.e. it can be used for defining relations among OWL classes, treating them as instances. The definition of the situation descriptors follows a tagging-like procedure, where the complex activities of the domain (designated through describes property assertions) are annotated with relevant observation classes (designated through dependency property assertions). For example, the situation descriptor of the tea drinking activity in the kitchen that is inferred on the basis of detecting a drinking event while the person is sitting in the table zone and uses a spoon and a tea cup can be defined as:

```
:drink_tea a sd:SituationDescriptor ;
```

```
sd:describes em:DrinkTea .
sd:dependency em:Sitting, em:TableZone,
   em:Spoon, em:TeaCup, em:Drink ;
```

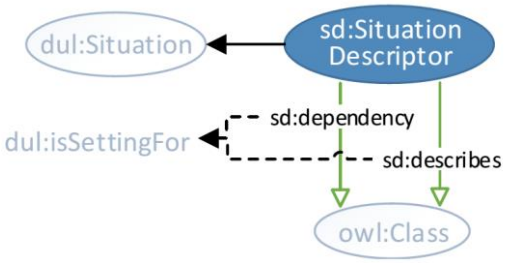


Figure 18. The Situation Descriptor model.

The motivation behind our decision to annotate activities with observation classes is two-fold. First, from a practical perspective, we believe that the resulting models are intuitive and can be easily defined, reused and extended in different domains. Second, from a theoretical perspective, the fact that classes can participate in instance-like property assertions enables the definition of contextualisations beyond standard OWL semantics. For example, OWL class semantics can model only domains where instances are connected in a tree-like manner. Therefore, it is not possible to model relations, e.g. temporal constraints, through TBox axioms, i.e. complex class descriptions, that involve classes not connected in a treelike manner. On the other hand, such arbitrary relations can be captured at the instance level, allowing additional descriptive context (views) to be assigned to the activity of situation descriptors.

Situation descriptor semantics. The situation descriptors encapsulate two semantic relations, namely descriptor unfolding and dependency linking, towards supporting knowledge sharing and reuse. Descriptor unfolding refers to the extension of the dependency set of descriptors taking into account activity hierarchies. Dependency linking is activated when the dependency set of a descriptor involves a complex activity, whose dependencies are also added to the initial dependency set.

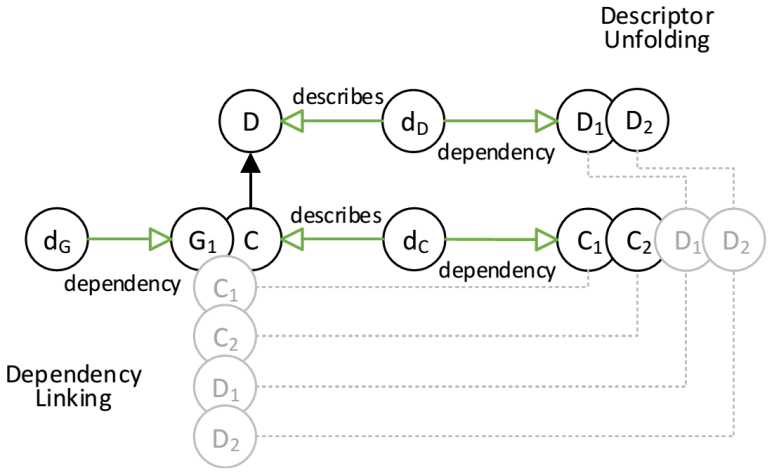


Figure 19. Visual explanation of dependency linking and descriptor unfolding semantics

Figure 19 graphically illustrates the above cases, where C and D are two activity classes ($C \sqsubseteq D$), and d_G, d_C and d_D are three situation descriptors with dependency sets $\{G_1, C\}$, $\{C_1, C_2\}$ and $\{D_1, D_2\}$, respectively. The d_C and d_D descriptors describe activity classes with subclass relations and therefore, the dependency set of d_C can be extended with the dependency set of d_D (descriptor unfolding). As far as the d_G's dependency set is concerned, it involves activity C, and therefore, it is extended with the dependency set of C's descriptor, namely d_C (dependency linking). Both relations are implemented in terms of OWL 2 property paths. Assuming that isDescribedBy is the inverse property of describes, the semantics of descriptor unfolding is given by (1), whereas (2) implements the notion of dependency linking.

$$describes \circ subClassOf \circ isDescribedBy \circ dependency \sqsubseteq dependency \quad (1)$$

$$dependency \circ isDescribedBy \circ dependency \sqsubseteq dependency \quad (2)$$

5.2 Interpretation Layer

The interpretation layer implements uses SPARQL CONSTRUCT graph patterns and procedural code to analyse observation traces and extract complex activities. Three steps are involved: (a) definition of local context, (b) identification of context connections and (c) recognition and classification of situations. In the following we briefly describe each phase, presenting also example SPARQL rules.

5.2.1 Local Contexts

Each observation is associated with a local context capturing information about the neighbouring observations and the most plausible complex activities the observation can be part of. Figure 20 presents the vocabulary for defining local contexts. The primary observation of the local context is designated by the withObservation property, whereas the neighbouring observations are captured using hasNeighbour property assertions. The most plausible activities of the primary observation are modelled using ClassificationEntity instances that encapsulate information relevant to the plausibility of the primary observation to be part of a complex activity and the complex activity itself (classifies property). The instantiation of local contexts is performed using the following SPARQL rule (Rule 1).

```
## Rule 1
CONSTRUCT {
  _:b0 a impl:LocalContext ;
    impl:withObservation ?this .
}
WHERE {
  ?this a em:Observation .
  FILTER NOT EXISTS {
    ?x impl:withObservation ?this .
  } .
}
```

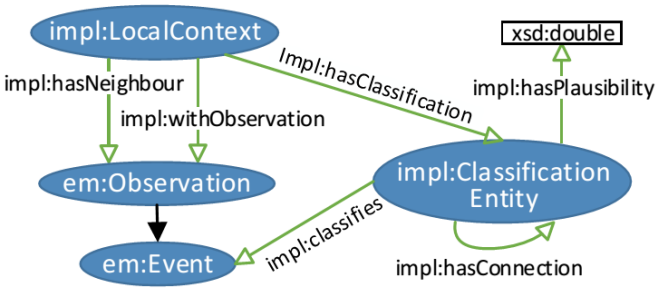


Figure 20. Local context model.

The association of local contexts with classification entities involves the combination of situation descriptor definitions. As Rule 2 depicts in the WHERE clause, this procedure involves (a) the extraction of the most specific class (?direct) of each primary observation ?o (using the :directClass SPIN function), and (b) the matching of situation descriptors (?descriptor) that contain in their dependency sets either the ?direct class or a subclass of ?direct. Upon the successful pattern matching of the graph, a classification entity is asserted that classifies the described complex activity ?c of the ?descriptor and is associated with the local context of the primary observation.

```
## Rule 2
CONSTRUCT {
  _:b0 a impl:ClassificationEntity ;
    impl:classifies ?c .
  ?this impl:hasClassification _:b0 .
}
WHERE {
  ?this impl:withObservation ?o .
  BIND (:directClass(?o) AS ?direct) .
  ?descriptor a sd:SituationDescriptor ;
    sd:describes ?c ;
    sd:dependency ?d .
  ?direct rdfs:subClassOf ?d .
  FILTER NOT EXISTS {
    ?this impl:hasClassification _:0 .
    _:0 impl:classifies ?c .
  } .
}
```

Finally, the plausibility of the primary observation to belong to each complex activity of the associated classification entities is given by the φ similarity (3). The similarity is computed between the set with the most specific neighbouring observation classes N_r and the dependency set of a context descriptor d as

$$\varphi(N^r, d) = \frac{\sum_{\forall n \in N^r} \max_{\forall c \in d} \left[\frac{|U(n) \cap U(c)|}{|U(n)|} \right]}{|N^r|} \quad (3)$$

where *U(C)* is the set of superclasses of C. As neighbours of a primary observation o we define the observations that either overlap with o or are the r-nearest to o, based on their temporal ordering. Intuitively, φ captures the local plausibility of an observation to be part of a complex activity. If $\varphi = 1$, then all the classes types of the neighbouring observations appear in some d and, therefore, it is very likely that the local context is part of the complex activity C.

5.2.2 Context Connections

The next step is to define context connections, that is, links among local contexts that will form the final situations for activity classification. More superficially, two local contexts are connected only if they share the same complex domain activity classification. The rationale behind context connections is that if two neighbouring local contexts have the same classification class, then it is very likely that the corresponding observations belong to the same complex activity. The following SPARQL rule implements the semantics of context connections.

```
## Rule 3
CONSTRUCT {
  ?li impl:hasConnection ?lj.
}
WHERE {
  ?li a impl:LocalContext;
    impl:withObservation ?oi;
    impl:hasClassification
      [impl:classifies ?Ci].
  ?lj a impl:LocalContext;
    impl:withObservation ?oj;
    impl:hasClassification
      [impl:classifies ?Ci].
  FILTER (?oi != ?oj).
  NOT EXISTS {?li impl:hasConnection ?lj}.
}
```

After the identification and assignment of context connections is complete, each local context is linked to any other relevant local context in the neighbour. This procedure is graphically illustrated in Figure 21, where the dots represent local contexts and the arrows correspond to context connections among local contexts with similar classification class C_i.

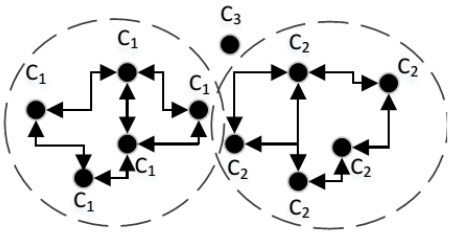


Figure 21 Example connections among local contexts

5.2.3 Classification of Situations

The situations represent meaningful subsets of observations that signify the presence of complex activities and are derived by traversing the context connection paths. Figure 21 visualises as dotted circles the two situations that are extracted for two complex activities C_1 and C_2. Figure 22 depicts the ontology vocabulary used for representing situations, whereas the following SPARQL rule collects observations from local contexts that classify the same complex activity as the situation instance, populating the dul:includesEvent property with observation instances.

```
CONSTRUCT {
  ?this dul:includesEvent ?o .
}
WHERE {
  ?this a impl:Situation;
    impl:interprets ?c .
  ?l a impl:LocalContext ;
    impl:hasClassification [impl:classifies ?c];
    impl:withObservation ?o .
}
```

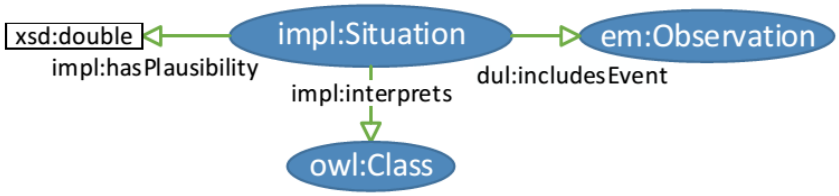


Figure 22. The Situation model for capturing the classification context of a group of observations

Having identified the situations, the last step of the algorithm is to compare the observation types of each situation against the observation types of the situation descriptors (:dependency property assertions) to retrieve the most plausible context. Similarly to φ in (3), the σ function defines the similarity of a situation descriptor's dependency set d_c against the set with the observation types *ObsT* of a situation:

$$\sigma(d_C, Obs_T) = \frac{\sum_{\forall n \in d_C} \max_{\forall c \in Obs_T} \left[\frac{|U(n) \cap U(c)|}{|U(c)|}\right]}{|d_C|} \quad (4)$$

If σ = 1, then all the classes in d_C appear in *ObsT*, meaning that the situation can be considered identical to the context descriptor d_C, and, therefore, to the complex class C. Otherwise, the context dependency d_C with the larger σ similarity is selected as the final complex activity classification of the situation.

5.3 Interleaved Activities

Several ontology-based reasoning architectures and prototypes have been proposed for activity recognition, each of which follows a different approach for handling intrinsic characteristics of the domain, such as data heterogeneity, temporal extension, noise, uncertainty and missing information. However, little focus has been given on the recognition of *interleaved activities* (i.e. non-consecutive), simplifying the problem of activity recognition

to only recognizing sequential activities, which is usually an unrealistic assumption. In real-world situations, activities may be performed in an interleaving manner, where one activity may be temporarily paused in order to perform one or more other activities. For example, an individual may be preparing a tea when the phone rings, so they have to pause the activity to answer the phone. Key challenges in this context involve the recognition of the start and end timestamps of all the activities involved and the derivation of the contextual interval when each activity was active, e.g. to classify interrupted instances of the same task as a single activity.

In the final version of the multi-parametric behaviour interpretation framework, we have investigated the use of *defeasible reasoning* [50] for detecting and classifying interleaved activities. Defeasible reasoning deploys a flexible conflict resolution framework for handling inconsistent and conflicting information, which is typical for (inherently uncertain and noisy) data coming from heterogeneous sensors. More specifically, we have defined a defeasible reasoning layer that can be used on top of existing ADL frameworks to facilitate the recognition of interleaved activities. The framework (ReDef) is based on the use of OWL 2 ontology models for capturing common sense knowledge regarding the context of the domain activities, and provides a set of defeasible rules that introduce semantic relationships among interleaved activities, such as telicity and contextual dependencies.

5.3.1 Defeasible Reasoning

Defeasible logics is a non-monotonic logics formalism that delivers intuitive knowledge representation and advanced conflict resolution mechanisms. In defeasible logics there are three distinct types of rules:

- *Strict rules* are denoted by $A \rightarrow p$ and are interpreted in the typical sense: whenever the premises are indisputable, then so is the conclusion.
- *Defeasible rules* are denoted by $A \Rightarrow p$ and, contrary to strict rules, they can be defeated by contrary evidence. Two defeater examples are r_1: *noon*(X) $\Rightarrow$ *havingLunch*(X), which reads as "*at noon, individual X* (i.e. the inhabitant of the house) *is probably having lunch*", and r_2: *onThePhone*(X) $\Rightarrow \neg$*havingLunch*(X), which reads as "*when X is on the phone then he/she is probably not having lunch*".
- *Defeaters* are denoted by $A \rightsquigarrow p$ and do not actively support conclusions, but can only prevent deriving some of them. In other words, they are used to defeat respective defeasible conclusions, by producing evidence to the contrary. A defeater example is: r_1': *sleep*(X) $\rightsquigarrow \neg$*havingLunch*(X) ("*when X is sleeping then he/she is definitely not having lunch*"), which can defeat e.g. rule r_1 mentioned previously.

Additionally, the *superiority relationship* is used for resolving conflicts among defeasible rules. For example, given the defeasible rules r_1 and r_2 above, no conclusive decision can be made about whether X is having lunch or not. But, if the superiority relationship $r_2 > r_1$ is introduced, then r_2 overrides r_1 and we can eventually conclude that the X is not having lunch after all. In this case rule r_2 is called *superior* to r_1 and r_1 *inferior* to r_2. Note that the relation $>$ is acyclic.

The advantages of applying defeasible instead of classical logics are outlined as follows:

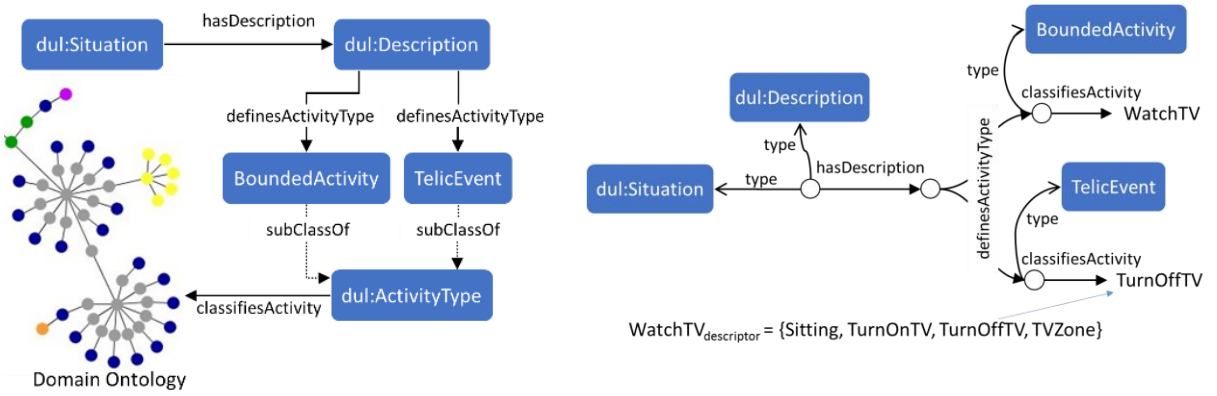
- Defeasible logics have low computational complexity;
- They allow for reasoning with incomplete information; this is a critical trait in sensor environments, where perfect knowledge of the environment is very hard, if not impossible, to achieve;

- They introduce non-monotonicity, which leads to a more intuitive type of reasoning, much closer to human reasoning, where the emergence of new information can lead to abandoning (i.e. defeating) previously established conclusions and adopting new ones.

5.3.2 Modelling Activity Telicity

ReDef provides two lightweight ontology patterns for capturing the concept of activity telicity, i.e. the context that designates that an activity has been completed. Both patterns implement the DnS ontology pattern of DUL and make use of the meta-modelling capabilities of OWL 2.

Telic event pattern. The telic event pattern enables to formally define the terminating state of a complex activity, i.e. the observation type that belongs to the activity's situation descriptor and denotes the completion of the activity. This pattern can be used for modelling telicity either for activities that do have endpoints, e.g. the event of turning off the TV can be considered as the telic event of watching TV. Figure 23 (a) depicts the schema of the telic event pattern, while Figure 23 (b) illustrates an example instantiation for modelling the telic event of watching TV. Following the conceptual model of DnS, the instantiation of the pattern involves the definition of a description instance that captures information about the activity type of interest and the telic event. The conceptual model of DnS also requires the assertion of a situation instance that references (via the `hasDescription` property assertion) the description instance. It is worth noting that the instantiation of the pattern involves the use of ontology classes in property assertions, e.g. in `defineActivityType`. The circles in Figure 23 (b) denote anonymous ontology instances that instantiate the pattern's concepts.



(a) (b)

Figure 23 (a) Telic event pattern; (b) Example instantiation for the `WatchTV` activity

Inter-context Telicity. While for some activities it is possible to select an observation from their situation descriptors to play the role of the telic event, there are other activities that cannot be bounded to specific endpoints. For example, preparing breakfast is a dynamic task that involves many activities without a predefined order or terminating contexts. For such activities, telicity cannot be defined by means of an observation that belongs to the situation descriptors.

In order to support the concept of telicity for activities that cannot be explicitly linked with a terminating state, ReDef provides the pattern depicted in Figure 24 (a). The idea behind this pattern is to capture activity telicity by means of existence of another context (inter-context telicity). For example, the detection of an activity relevant to cleaning the table in the morning

is an indication that the individual may have prepared a breakfast earlier, which can be considered as completed. Similar to the telic event pattern, the instantiation of this pattern requires the assertion of situation and description instances, designating the role of each instance by assigning it to the available concepts (`BoundedActivity` or `TelicContext`). Moreover, this pattern allows us to capture temporal dependencies among the bounded activities and the respective contexts. For example, the instantiation of the pattern in Figure 24 (b) explicitly models that the cleaning table context should follow the prepare breakfast activity.

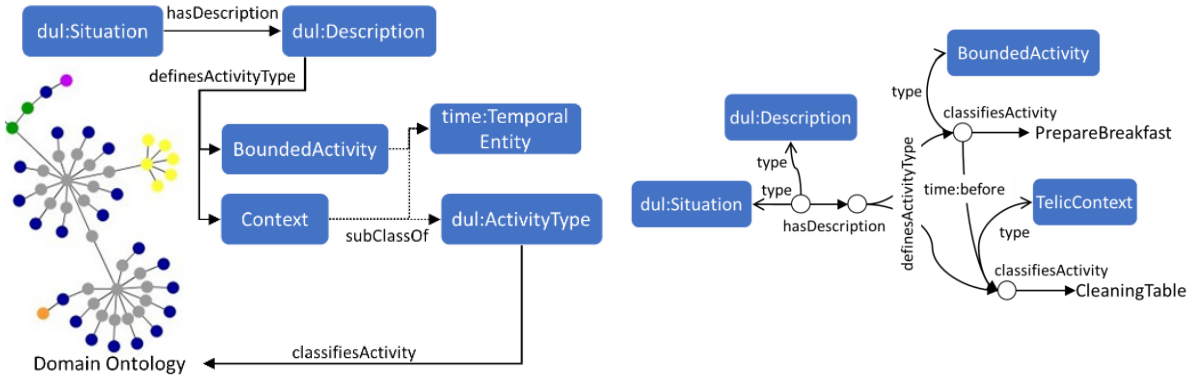


Figure 24 (a) Inter-context telicity pattern; (b) Example instantiation for `PrepareBreakfast`

5.3.3 Recognising Interleaved Activities

The aim of ReDef is to provide a framework that can be used on top of existing activity recognition solutions in order to enhance their performance with respect to the detection of interleaved activities. This is achieved by examining the already detected activities and their constituent observations to detect situations when the telicity patterns are satisfied in order to further aggregate the individual activities and derive interleaved tasks. As such, ReDef requires as input the following information:

- **Activity traces:** set of detected complex activities with start/end timestamps.
- **Sub-events:** the constituent parts (observations) of the complex activities.
- **Activity telicity patterns:** instantiations of the patterns described in section 5.3.2.

In the following, we assume that the rule-based methodology for determining which activities are interleaved is based on the following set of core predicates:

- `activity(A,T1,T2)`: `A` is an activity starting at `T1` and ending at `T2`.
- `type(A,P)`: Resource (observation/activity) `A` is of activity type `P`.
- `subEvent(O,A)`: Observation `O` belongs to activity `A`.

Interleaved Activities Through Direct Telicity. The following set of rules implements the semantics of the telic event pattern described in section 5.3.2, asserting pairs of interleaved activities. In addition to the core predicates, the predicate `telic(TL,A)` is defined that denotes that `TL` is the telic event for activity `A`.

r_1: activity(A1,T11,T12), activity(A2,T21,T22), T21 > T12, type(A1,A), type(A2,A), telic(TL,A), subEvent(Z,A2), type(Z,TL)

⇒ interleaved(A1,A2)

r_2: activity(A1,T11,T12), activity(A2,T21,T22), T21 > T12, type(A1,A), type(A2,A), telic(TL,A), subEvent(Z,A1), type(Z,TL)

⇒ ¬interleaved(A1,A2)

r_3: activity(A1,T11,T12), activity(A2,T21,T22), activity(A3,T31,T32), T21 > T12, T31 > T22, type(A1,A), type(A2,A), type(A3,A), telic(TL,A), subEvent(Z1,A2), subEvent(Z2,A3), type(Z1,TL), type(Z2,TL)

⇒ ¬interleaved(A1,A3)

r_2, r_3 > r_1

More specifically, rule r_1 determines when two separate activities constitute a single, interleaved one, based on the existence of the corresponding telic observation in the activity context that takes place last. On the other hand, rule r_2 establishes an exception to r_1 that takes place when the first activity (also) includes a telic observation. An additional exception, r_3, ensures that an activity is linked only with the most recent telic context. Consequently, these exceptions are introduced as superior to r_1 via the superiority relationship. When the execution of rules terminates, the pair of intervened activities are traversed to select the one with the longest duration as the final activity.

Interleaved Activities through Inter-context Telicity. In order to implement the semantics of the inter-context telicity pattern, the `telic` predicate is replaced by predicate `final(A)` indicating that activity `A` is completed (no subsequent activities of the same type may be appended to `A`). The following rule determines the `final` activities:

r_4: activity(A1,T11,T12), activity(B1,T21,T22), latest(A1,B1), type(A1,A), type(B1,B), telicContext(A,B)

⇒ final(A1)

where [latest(A1,B1), type(A1,A), type(B1, B)] retrieves the closest most recent activity of type `A` to type `B`.

Having detected the `final` activities, a rule set similar to the one presented in the previous subsection (rules r_2-r_3) has to be deployed, where the `telic` predicate is substituted by `final`.

5.4 Evaluation

The ontology-based fusion and activity recognition capabilities have been evaluated both in lab and home environments. For both environments, ground truth has been obtained through annotation, using a dedicated tool in the framework, based on images from ambient cameras.

We use the True Positive Rate (TPR) and Positive Predicted Value (PPV) measures, which denote recall and precision, respectively, to evaluate the performance. These measures are defined as:

$$TPR = \frac{TP}{TP + FN}, \; PPV = \frac{TP}{TP + FP}$$

where True Positives (TP) is the number of IADLs correctly recognized, False Positives (FP) is the number of IADLs incorrectly recognized as performed and False Negatives (FN) is the number of IADLs that have not been recognized.

5.4.1 Lab environment

The first step towards configuring the framework for the lab evaluation was to define the context dependency models for the 4 activities involved in the (short) experimentation protocol. This translates to recognizing the behavior of the participant with respect to the protocol specifications, i.e. recognizing the activities performed. Consequently, the aim is to augment the available set of observations of the deployment by extracting the pieces of aggregated information that cannot be provided by means of the installation sensors and components alone.

The context dependency models were derived after several iterations with the clinicians in order to clearly define when an activity should be considered successful or not in the experimentation context. For example, in order to recognize that the participant has answered the phone, clinicians suggested that the constituent activities "phone moved", "phone zone", "phone object" and "talking" need to be recognized by the components and fused by the high-level interpretation module. Based on the elicited knowledge, we defined the 4 context dependency models depicted in Table 7. The dependency models involve ontology concepts relevant to detected scene objects (from video analysis), the location of the participant (from wearable camera, ambient camera and presence sensors), the objects used by the participant (from tag sensors) and posture information (from video analysis). It should be noted that the experiment involves complementary modalities. For example, information about the location of the person is provided by three sensors (two cameras and a presence sensor).

Table 7 Context dependency models for the lab evaluation

Activity Concept	Context dependency set
PrepareDrugBox	DrugBoxMoved, DrugZone, DrugBoxObject, PillBoxMoved, PillBoxObject
AnswerPhone	PhoneMoved. PhoneZone, PhoneObject, Talking
EstaclishAccountBalance	Sitting, AccountZone, AccountMoved, AccountObject, PenObject
PrepareHotTea	KettleOn, TeaZone, KettleObject, TeaBagMoved, TeaBagObject, CupObject, CupMoved

Table 8 summarizes the performance of the framework on a dataset of 97 participants. The framework recognizes 3 out of 4 activities with an average recall and precision close to 82%, while the performance in recognizing the `EstablishAccountBalance` activity is quite low. This is mainly due to low classification performance demonstrated by the low-level components that detects relevant observations to this activity.

Table 8 Precision and recall for activity recognition in lab

Activity	Recall (TPR)	Precision (PPV)
PrepareDrugBox	0.833	0.813

PrepareHotTea	0.816	0.796
EstablishAccountBalance	0.292	0.393
AnswerPhone	0.827	0.814

In order to examine the noise tolerance properties of high-level interpretation with respect to the classification accuracy of the low-level components, we conducted a number of experiments combining different modalities. Although this is still a work in progress, the results so far suggest that the fusion of different modalities does not always improve the overall activity recognition performance. Instead, a flexible and dynamic integration solution is needed able to take into consideration the individual characteristics and capabilities of each component in diverse deployments. For example, Figure 25 and Figure 26 depict the recall and precision respectively of three configurations: HAR that performs activity recognition based on video analytics, SI that corresponds to the fusion of all the available modalities in the lab apart from HAR, and SI+HAR that denotes the combination of both. It is evident from the results that the overall performance (SI+HAR) is improved only when HAR achieves precision and recall close to 0.5 and above. In any other case, the incorporation of results from video analytics has a negative impact on the performance. For example, the recall and precision for the `EstablishAccountBalance` activity is reduced by 63% and 48%, respectively.

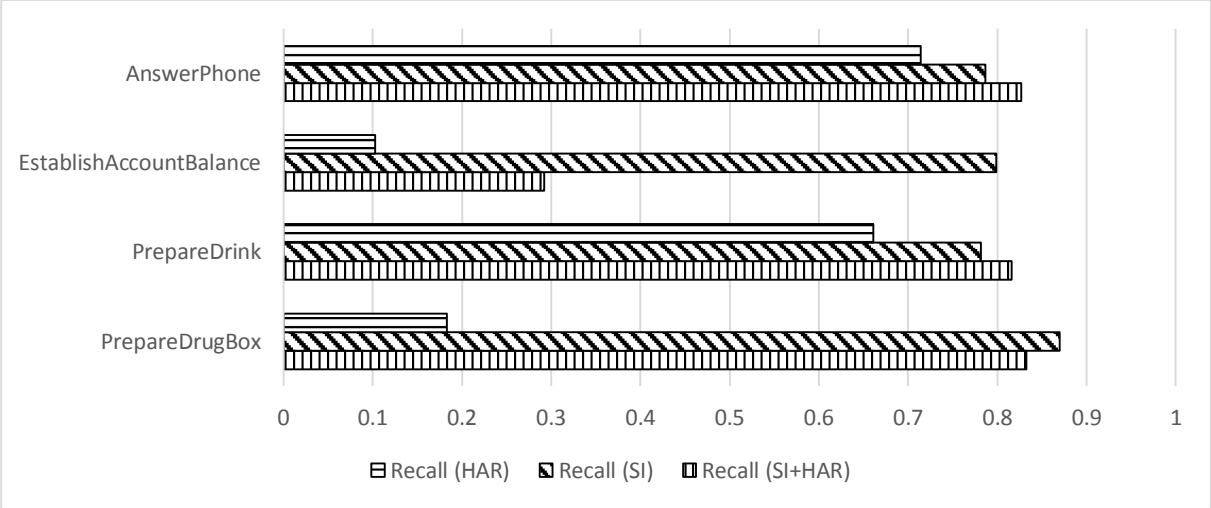


Figure 25 Recall of HAR, SI and their combination

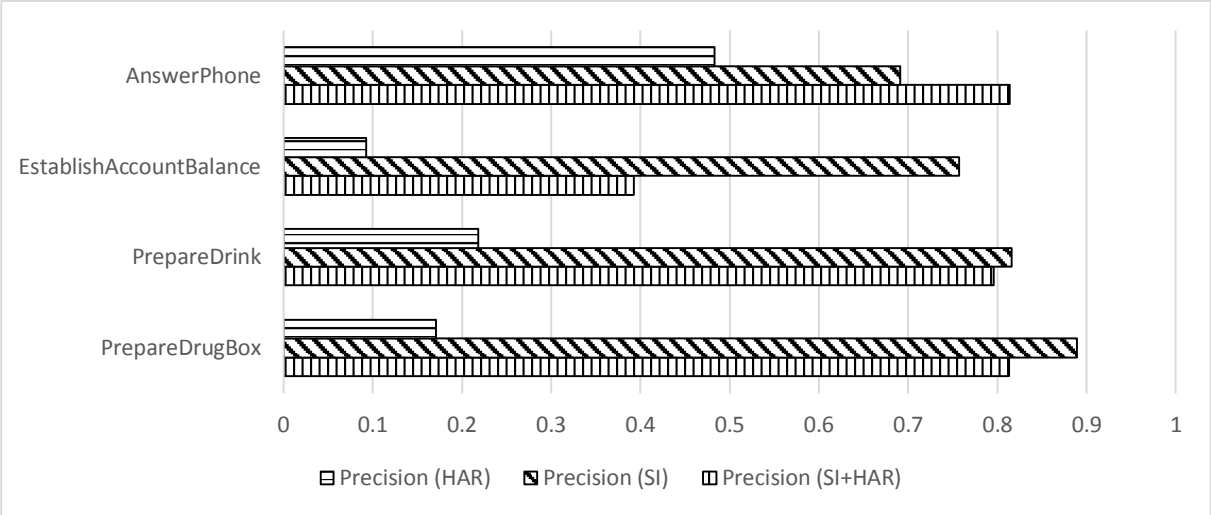


Figure 26 Precision of HAR, SI and their combination

5.4.2 Home environment

The aim of semantic interpretation in home is to detect daily activities performed by the individuals and assess their behavior. To this end, the configuration of the framework for the home deployment involved the definition of the context dependency models pertinent to the activities of interest. Annotation can be performed only for the activities taking place in the kitchen, where an ambient camera is available. The clinical expert suggested the monitoring of three activities in this room, namely making tea, cooking and drug box preparation. Table 9 depicts the pertinent context dependency models defined.

Table 9 Context dependency models for the home evaluation

Activity Concept	Context dependency set
MakeHotTea	KettleOn, TeaZone, KettleObject, TeaBagMoved, TeaBagObject, CupObject, CupMoved
Cooking	CookerOn, CookerZone, CutleryObjects
PrepareDrugBox	DrugBoxMoved, DrugZone, DrugBoxObject, PillBoxMoved, PillBoxObject, DrugCabinetMoved

ADL activity recognition performance has been evaluated on a dataset of 31 days (July 2015). The results are depicted in Table 10. We observe that the activities `PrepareDrugBox` and `PrepareHotTea` are recognized correctly in most cases. However, the recognition of cooking is a more challenging task, since it is usually being performed in an interleaved manner.

Table 10 Precision and recall for activity recognition in home

Activity	Recall (TPR)	Precision (PPV)
PrepareDrugBox	0.86	0.89
Cooking	0.61	0.68
PrepareHotTea	0.81	0.86

In the lab environment, participants perform the activities mostly sequentially. However, the assumption that individuals carry out a single activity each time falls short in more realistic environments, such as in home. In real-world situations, activities may be performed in an interleaving manner, where one activity may be temporarily paused in order to perform one or more other activities. For example, an individual may be preparing a tea when the phone rings, so they have to pause the activity to answer the phone. Similarly, in most cases individuals performs other activities while cooking, e.g. cleaning the table or vacuuming. In these cases, the interleaved contexts are recognized by our algorithms as individual activities, affecting the performance. Key challenges in this context involve the recognition of the start and end timestamps of all the activities involved and the derivation of the contextual interval when each activity was active, e.g. to classify interrupted instances of the same task as a single activity. We are currently investigating non-monotonic reasoning solutions to the aforementioned problem, investigating a combination of defeasible logic to handle conflicts and classify interrupted instances of the same task as a single activity.

5.4.3 Clinical Assessment, Monitoring and Intervention

All in all, the practical aim of WP5 is to collect and further analyse multi-modal results of the various components of the framework, so as to help stakeholders draw high-level conclusions regarding the behaviour and health status of the individuals being monitored. As such, caregivers and clinical experts are able to monitor the progress of individuals through personalised and integrated views on their condition, as well as they acquire objective observations that can be used for clinical assessment, design and update of interventions. To this end, several fusion algorithms and frameworks have been developed and integrated within WP5, able to combine heterogeneous pieces of information. Each modality generates information from a different perspective, but by combining them together we are in a position to infer far more about a person's activities than by any sensor alone. In addition, novel knowledge-driven frameworks have been proposed, such as the one described in this section, aiming to semantically interpret the aggregated data, detecting patterns, correlations among modalities and problematic situations that are of clinical interest. In close cooperation with WP6, intelligent feedback and alert services have been developed on top of WP5's knowledge base, highlighting to the end users critical situations or behavioural changes that need further investigation.

The performance of WP5's modules has been evaluated throughout the course of the project, in different settings and with different aims. More specifically, in the Lab setting, WP5's fusion algorithms derive information about the performed activities, such as the start time or duration, while high-level interpretation modules detect problematic situations, capturing clinical knowledge about lab-related issues, such as missed or repeated activities. The multi-sensor analysis results of WP5, and more specifically, of the Semantic Interpretation (SI) component, are used to automatically classify Lab participants in different groups. As described, demonstrated and clinically validated in D8.5, the Dem@Care system can improve the early detection of dementia, obtaining high accuracy rates in differentiating between healthy, MCI and AD subjects using the sensor data extracted. For example, in the Thessaloniki Lab long protocol, 60 patients were split in 45 training and 15 test sets. Labelled data were used to train a SVM classifier and produce a model, while test data were used to predict and evaluate the overall system. 16 attributes were used (time duration and successful attempts of each activity from the SI analysis). This resulted in a highly accurate SVM model

that achieved 89.15±0.20% mean average accuracy on itself, while test data were predicted with 65.22±0.66% accuracy rate.

In Home and Nursing Home settings, WP5 affords automated reasoning mechanisms for the high-level interpretation of the PwD behaviour via the integration and semantic fusion of the information made available through monitoring, as well as mechanisms for dynamic patient profiling so as to endow behaviour interpretation reasoning with personalisation capabilities. Specifically in Nursing Home, the primary goal is to promote enablement and safety of the PwD via appropriate feedback to the PwD and the attending clinician(s). To this end, through intelligent monitoring and fusion, WP5's decision making mechanisms derive daily activity logs, while the periodic extraction of habitual patterns enables the detection of deviations from normal behaviours and correlations among modalities. The performance of the modules have been evaluated both experimentally, as presented in the respective technical deliverables, and qualitatively, as described in the Final Pilot Evaluation Report (D8.5), through clinical assessments and selected interventions.

6 Ontologies

Based on the requirements set by WP2, the dependencies incurring from the interaction with the other WPs and the reasoning requirements in WP5, the Dem@Care ontology has been developed over the course of the project that provides the necessary knowledge structures and vocabularies for representing data and knowledge in different levels of abstraction. The Dem@Care ontology mainly comprises four modules: i) the lab ontology that formalises information relevant to the ecological assessment taking place in the laboratory environment, ii) the home/nursing home ontology that formalises information relevant to the monitoring of PwD in home and nursing homes environments, such as events, problems, entities (e.g. objects, places), iii) the questionnaire ontology that allows for modelling questionnaire-related information and iv) the context descriptor ontology that is used for defining the semantic of higher level activities and knowledge patterns. Most of the ontologies have been formalised using the DnS (Descriptions and Situations) pattern of the DOLCE+DnS Ultralite ontology.

In an effort towards standardisation, we submitted in 2014 the Dem@Care ontology for the Lab setting[1] to the LOV (Linked Open Vocabularies) community[2]. LOV provides a growing ecosystem of linked open vocabularies (RDFS and/or OWL ontologies) used in the Linked Data Cloud. The submitted ontologies can be queried either at vocabulary level or at element level, exploring the vocabulary content using full-text faceted search, and finding metrics about the use of vocabularies in the Semantic Web. By submitting the Dem@Care Lab ontology to the LOV community (Figure 27), we enabled its reuse by other datasets in the Linked Data Cloud, promoting at the same time interoperability and knowledge sharing. A human-readable description of the vocabulary is also available[3].

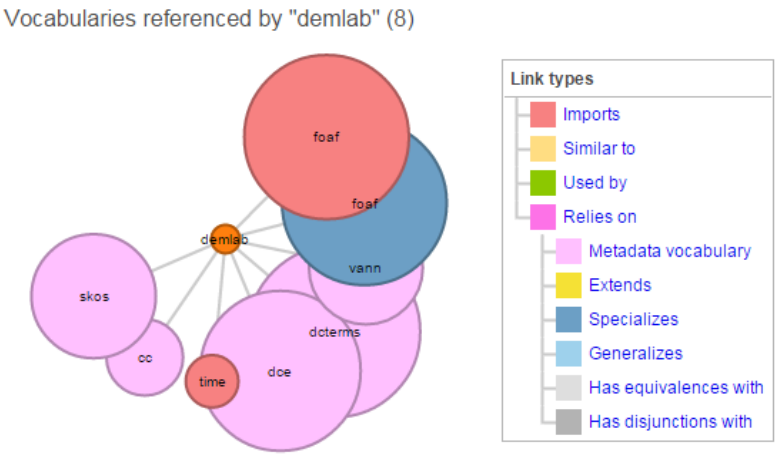


Figure 27. The Dem@Care Lab Ontology at LOV

The standardisation efforts have been continued during the fourth year, publishing to the LOV community the Domain Context Descriptor ontology (Figure 28), which has been developed

[1] http://lov.okfn.org/dataset/lov/details/vocabulary_demlab.html

[2] http://lov.okfn.org/dataset/lov/

[3] http://www.dem-care.eu/ontologies/demlab.html

for formally describing the high-level context pertinent to ADLs. A human-readable description of the vocabulary is also available[4].

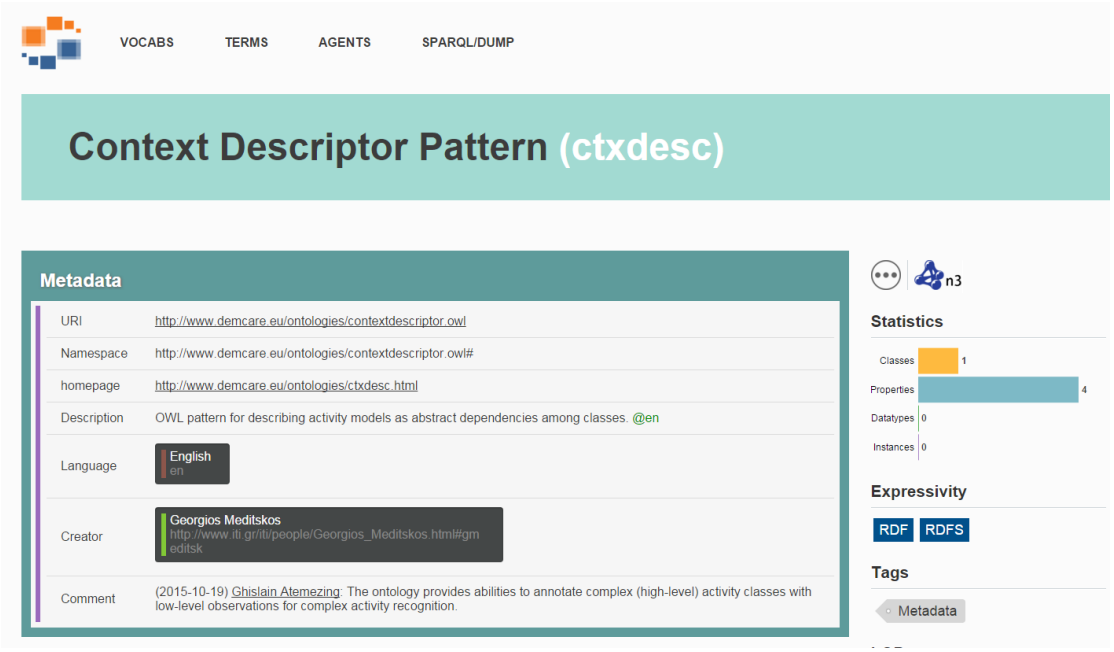


Figure 28. The Domain Context Descriptor ontology at LOV

The home/nursing home (Figure 30) and questionnaire (Figure 29) ontologies have been also submitted to the community and they are under revision.

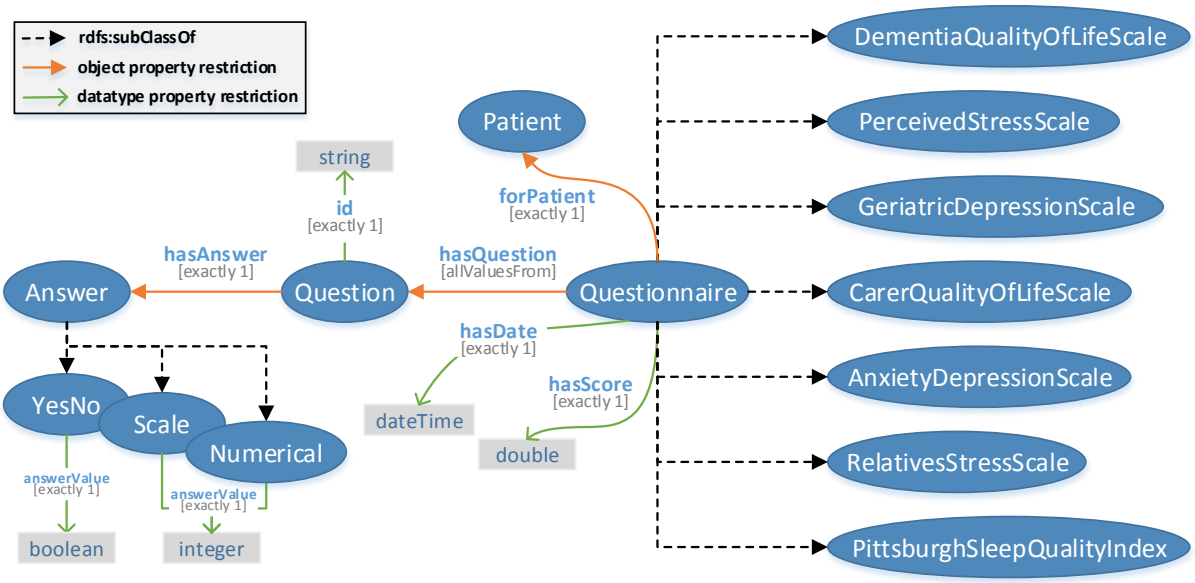


Figure 29. The Questionnaire Ontology

[4] http://www.demcare.eu/ontologies/ctxdesc.html

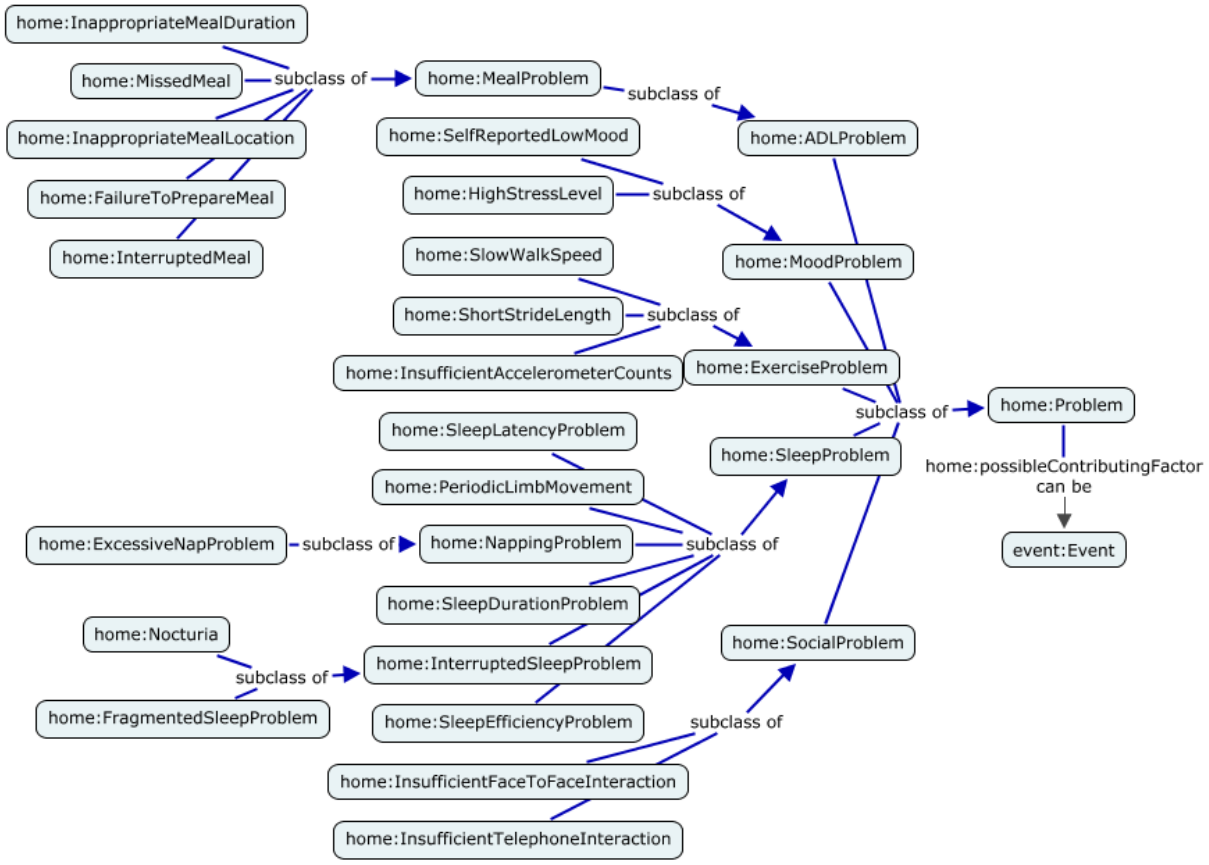


Figure 30. Excerpt from the Problem class hierarchy

6.1 Metrics

In this section, we present some detailed metrics about the Dem@Care ontologies. One of the crucial issues in ontology evaluation is the identification of anomalies or worst practices in ontology development. In [58] the authors describe a set of common errors made by developers and knowledge engineers during ontology modelling. Moreover, in [29] a classification of errors identified during the evaluation of consistency, completeness, and conciseness of ontology taxonomies is provided. Finally, in [55] authors identify an initial catalogue of common pitfalls.

OOPS! [56] is a tool that scans ontologies looking for potential pitfalls that could led to modeling errors. Its main functionality is to analyze ontologies via URL or RDF coding and to inform developers about which elements of the ontology are possibly affected by pitfalls or syntax errors. It also provides modelling suggestions for some relationships.

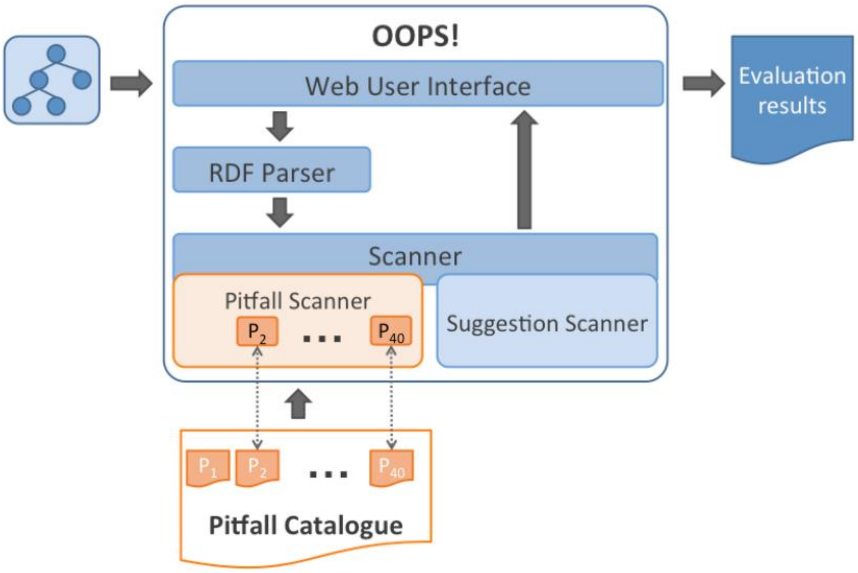


Figure 31. OOPS! architecture - http://www.oeg-upm.net/oops

Currently, 32 pitfalls have been implemented [55] and the detection is automated in 3 ways:

- Lexical content analysis: make use of the content of annotations and identifiers for detecting pitfalls, e.g. P22: Using different naming criteria in the ontology.
- Seeking a particular characteristic: check general characteristics of the ontology not related to the internal structure of the ontology or to the content of the lexical entities, e.g. P36. URI contains file extension.
- Structural pattern: analyze the internal structure of the ontology, seeking specific parts of the model, e.g. P5: Defining wrong inverse relationships

The tool supports the following evaluation categories:

- **Structural Dimension**
  - Modelling Decisions: Checks for pitfalls P02, P03, P07, P21 and P24.
  - Wrong Inference: Checks for pitfalls P05, P06 and P19.
  - No Inference: Checks for pitfalls P11, P12 and P13.
  - Real World Modelling or Common Sense: Checks for pitfall P10.
  - Ontology language: Checks for pitfalls P34, P35 and P38.
- **Functional Dimension**
  - Requirements Completeness: Checks for pitfall P04.
  - Application context: Checks for pitfalls P36, P37, P38, P39 and P40.
- **Usability-Profiling Dimension**
  - Ontology Clarity: Checks for pitfalls P08 and P22.
  - Ontology Understanding: Checks for pitfalls P02, P07, P08, P11, P12, P13 and P20.
- **Consistency**
  - For this evaluation criteria the following pitfalls will be checked: P05, P06, P07, P19 and P24.
- **Completeness**

- For this evaluation criteria the following pitfalls will be checked: P04, P10, P11, P12 and P13.

- **Conciseness**
  - For this evaluation criteria the following pitfalls will be checked: P02, P03 and P21.

In the following describe the use of OOPS! to scan for pitfalls the Dem@Care ontologies with respect to the aforementioned evaluation categories supported by the tool.

6.1.1 Lab ontology

After analyzing the lab ontology with OOPS!, we got 14 pitfalls and 1 suggestion. The results are depicted in Table 11. As it can be observed, the majority of the pitfalls are relevant to external/imported constructs, such as concepts from the FOAF ontology. For example, the evaluation process has identified a potential pitfall regarding the definition of wrong equivalent relationships (P27) between the creator and maker properties of FOAF and DC TERMS, or the recursive definition (P24) of OWL Time properties. However, there are some pitfalls relevant to the Lab ontology, such as P07 regarding the definition of concepts that merge two or more individual concepts, P08 about missing annotations or missing domain/range restrictions (P11). The majority of the identified pitfalls have been fixed in the final version of the Lab ontology. P36 has not been fixed, since the use of a file extension in the URI does not affect the visibility of the ontology. Table 12 depicts various ontology metrics as these have been calculated by Protégé[5] and OntoMetricCalc[6] Java API.

Table 11 OOPS! evaluation results for the Lab ontology

Code	Name	Description	Importance	#Affected elements
P04	Creating unconnected ontology elements	Ontology elements (classes, relationships or attributes) are created with no relation to the rest of the ontology. An example of this type of pitfall is to create the relationship "memberOfTeam" and to miss the class representing teams; thus, the relationship created is isolated in the ontology.	Minor	http://xmlns.com/foaf/0.1/Project http://purl.org/dc/dcam/VocabularyEncodingScheme http://xmlns.com/foaf/0.1/LabelProperty
P07	Merging different concepts in the same class	A class is created whose identifier is referring to two or more different concepts. An example of this type of pitfall is to create the class "StyleAndPeriod", or "ProductOrService".	Minor	http://www.demcare.eu/ontologies/demlab.owl#NeuropsychiatricAndMoodAssessment http://purl.org/dc/terms/LocationPeriodOrJurisdiction http://purl.org/dc/terms/MediaTypeOrExtent http://purl.org/dc/terms/SizeOrDuration

[5] protegewiki.stanford.edu

[6] http://sourceforge.net/projects/ontometricalc/

P08	Missing annotations	Ontology terms lack annotations properties. This kind of properties improves the ontology understanding and usability from a user point of view.	Minor	http://www.demcare.eu/ontologies/demlab.owl#Participant http://www.demcare.eu/ontologies/demlab.owl#DirectedDiscussionTask http://www.demcare.eu/ontologies/demlab.owl#DiagnosisType **+ 149 more**
P11	Missing domain or range in properties	Relationships and/or attributes without domain or range (or none of them) are included in the ontology. There are situations in which the relation is very general and the range should be the most general concept "Thing". However, in other cases, the relations are more specific and it could be a good practice to specify its domain and/or range. An example of this type of pitfall is to create the relationship "hasWritten" in an ontology about art in which the relationship domain should be "Writer" and the relationship range should be "LiteraryWork".	Important	http://www.demcare.eu/ontologies/demlab.owl#participates http://www.demcare.eu/ontologies/demlab.owl#isClinicalRecordOf http://www.demcare.eu/ontologies/demlab.owl#hasUPDRSScore **+72 more**
P12	Missing equivalent properties	When an ontology is imported into another, classes that are duplicated in both ontologies are normally defined as equivalent classes. However, the ontology developer misses the definition of equivalent properties in those cases of duplicated relationships and attributes. For example, the classes "CITY" and "City" in two different ontologies are defined as equivalent classes; however, relationships "hasMember" and "has-Member" in two different ontologies are not defined as equivalent relations.	Important	http://xmlns.com/foaf/0.1/title http://purl.org/dc/terms/title http://xmlns.com/foaf/0.1/givenname http://xmlns.com/foaf/0.1/givenName http://xmlns.com/foaf/0.1/family_name **http://xmlns.com/foaf/0.1/familyName**
P13	Missing inverse relationships	This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".	Minor	**(might be inverse)** http://xmlns.com/foaf/0.1/theme http://xmlns.com/foaf/0.1/fundedBy http://xmlns.com/foaf/0.1/fundedBy http://xmlns.com/foaf/0.1/logo **(no inverse suggestion)** http://www.demcare.eu/ontologies/demlab.owl#

				measuredData http://www.demcare.eu/ontologies/demlab.owl#handTrajectory **+60 more**
P22	Using different naming criteria in the ontology	Ontology elements are not named using the same convention within the whole ontology. It is considered a good practice that the rules and style of lexical encoding for naming the different ontology elements is homogeneous within the ontology. One possibility for rules is that concept names start with capital letters and property names start with non-capital letters. In the case of style, there are different options such as camel case, hyphen style, underscore style, and the combinations.	Minor	http://xmlns.com/foaf/0.1/topic_interest http://xmlns.com/foaf/0.1/workplaceHomepage
P24	Using recursive definition	An ontology element is used in its own definition. For example, it is used to create the relationship "hasFork" and to establish as its range the following "The set of restaurants that have at least one value for the relationship "hasFork".	Important	http://www.w3.org/2006/time#Instant http://www.w3.org/2006/time#Interval
P27	Defining wrong equivalent relationships	Two relationships are defined as equivalent relations when they are not necessarily.	Critical	http://purl.org/dc/terms/creator http://xmlns.com/foaf/0.1/maker
P30	Missing equivalent classes	When an ontology is imported into another, classes with the same conceptual meaning that are duplicated in both ontologies should be defined as equivalent classes to benefit the interoperability between both ontologies. However, the ontology developer misses the definition of equivalent classes in the cases of duplicated concepts. An example of this pitfall can be not to have the equivalent knowledge explicitly defined between "Trainer" (class in the imported ontology) and "Coach" (class in the ontology about sports being developed).	Important	**(might be equivalent classes)** http://www.w3.org/2006/time#Year http://www.w3.org/2000/01/rdf-schema#Class http://www.demcare.eu/ontologies/demlab.owl#Task http://xmlns.com/foaf/0.1/Project
P31	Defining wrong equivalent classes	Two classes are defined as equivalent when they are not necessarily equivalent. For example, defining Car as equivalent to Vehicle	Critical	**(might not be equivalent classes)** http://xmlns.com/foaf/0.1/Image http://schema.org/ImageObj

				ect http://schema.org/CreativeWork http://xmlns.com/foaf/0.1/Document
P34	Untyped class	A resource is used as a class, e.g. appearing as the object of an "rdf:type", "rdfs:domain", or "rdfs:range" statement, or as the subject or object of an "rdfs:subClassOf" statement, without having been declared as a Class.	Important	http://schema.org/ImageObject http://schema.org/CreativeWork http://schema.org/Person http://www.w3.org/2000/10/swap/pim/contact#Person
P36	URI contains file extension	Guidelines suggest avoiding file extension in persistent URIs, particularly those related to the technology used, as for example ".php" or ".py". In our case we have adapted it to the ontology web languages used to formalized ontologies and their serializations. In this regard, we consider as pitfall including file extensions as ".owl", ".rdf", ".ttl", ".n3" and ".rdfxml" in an ontology URI. An example of this pitfall (at 29th June, 2012) could be found in the "BioPAX Level 3 ontology (biopax)" ontologyOs URI (http://www.biopax.org/release/biopax-level3.owl) that contains the extension ".owl" related to the technology used.	Minor	http://www.demcare.eu/ontologies/demlab.owl
P40	Namespace hijacking	This means reusing or referring to terms from other namespaces not actually defined in such namespace. This pitfall is related to the Linked Data publishing guidelines, "Only define new terms in a namespace that you control". Example: the "WSMO-Lite Ontology (wl)" which URI is http://www.wsmo.org/ns/wsmo-lite#, uses http://www.w3.org/2000/01/rdf-schema#Property that is not defined in the rdf namespace (http://www.w3.org/2000/01/rdf-schema#) instead of using http://www.w3.org/1999/02/22-rdf-syntax-ns#Property, that is ac-tually defined in the rdfs namespace (http://www.w3.org/1999/02/22-rdf-syntax-ns#).	Critical	http://creativecommons.org/ns#license
	SUGGESTION: symmetric or transitive object	The domain and range axioms are equal for each of the following object properties. Could they be symmetric or		http://xmlns.com/foaf/0.1/theme http://xmlns.com/foaf/0.1/th

	properties.	transitive?		umbnail http://xmlns.com/foaf/0.1/fundedBy http://xmlns.com/foaf/0.1/knows http://xmlns.com/foaf/0.1/logo http://xmlns.com/foaf/0.1/based_near http://www.w3.org/2006/time#intervalOverlaps http://www.w3.org/2006/time#intervalStarts http://www.w3.org/2006/time#intervalEquals http://www.w3.org/2006/time#intervalDuring http://www.w3.org/2006/time#intervalBefore http://www.w3.org/2006/time#intervalMeets http://www.w3.org/2006/time#intervalFinishes

Table 12 Various ontology metrics calculated for the Lab ontology

DL expressivity	SHOIG(D)
Axioms	2111
Logical axiom count	538
Number of Classes	220
Number of Internal Classes	0
Number of Named Classes	104
Number of Anonymous Classes	116
Number of Restrictions	94
Number of Complement Classes	0
Number of Enumerated Classes	7
Number of Intersection Classes	6
Number of Union Classes	12
Number of Leaf Classes	110
Number of Root Classes	146
Maximum Depth of Class Inheritance Tree	3
Average Depth of Class Inheritance Tree	0.99

Maximum Number of Class Ancestors	4
Average Number of Class Ancestors	1.04
Number of Properties	218
Number of Internal Properties	0
Number of Object Properties	94
Number of Functional Properties	1
Number of Inverse Functional Properties	5
Number of Symmetric Properties	0
Number of Inverse Properties	16
Number of Transitive Properties	1
Number of Datatype Properties	84
Number of Annotation Properties	72
Number of Individuals	45
Number of Internal Individuals	0
Number of Named Individuals	45
Number of Anonymous Individuals	0

6.1.2 Context Descriptor Ontology

OOPS! identified 16 pitfalls and 1 suggestion for the context descriptor ontology. The results are depicted in Table 13. Similarly to the DemLab ontology, the majority of the pitfalls are relevant to external vocabularies, such as P31 and P34. On the other hand, P12 was a quite useful finding, since the tool identified similarities between the concepts (a) dul:describes and ctxdesc:describes and (b) ctxdesc:isDescribedBy and dul:isDescribedBy, helping us to introduce new equivalent class axioms. Table 14 depicts various ontology metrics as these have been calculated by Protégé and OntoMetricCalc Java API.

Table 13 OOPS! evaluation results for the Context Descriptor ontology

Code	Name	Description	Importance	#Affected elements
P04	Creating unconnected ontology elements	Ontology elements (classes, relationships or attributes) are created with no relation to the rest of the ontology. An example of this type of pitfall is to create the relationship "memberOfTeam" and to miss the class representing teams; thus, the relationship created is isolated in the ontology.	Minor	http://xmlns.com/foaf/0.1/Project http://xmlns.com/foaf/0.1/LabelProperty
P08	Missing annotations	Ontology terms lack annotations properties. This kind of properties improves the ontology understanding and usability from a user point of view.	Minor	http://www.loa-cnr.it/ontologies/DUL.owl#DesignedSubstance http://www.demcare.eu/ontologies/contextdescriptor.owl#isDescribedBy

				http://www.loa-cnr.it/ontologies/DUL.owl#isEventIncludedIn **+21 more**
P11	Missing domain or range in properties	Relationships and/or attributes without domain or range (or none of them) are included in the ontology. There are situations in which the relation is very general and the range should be the most general concept "Thing". However, in other cases, the relations are more specific and it could be a good practice to specify its domain and/or range. An example of this type of pitfall is to create the relationship "hasWritten" in an ontology about art in which the relationship domain should be "Writer" and the relationship range should be "LiteraryWork".	Important	http://www.demcare.eu/ontologies/contextdescriptor.owl#isDescribedBy http://www.loa-cnr.it/ontologies/DUL.owl#directlyFollows http://xmlns.com/foaf/0.1/givenName **+11 more**
P12	Missing equivalent properties	When an ontology is imported into another, classes that are duplicated in both ontologies are normally defined as equivalent classes. However, the ontology developer misses the definition of equivalent properties in those cases of duplicated relationships and attributes. For example, the classes "CITY" and "City" in two different ontologies are defined as equivalent classes; however, relationships "hasMember" and "has-Member" in two different ontologies are not defined as equivalent relations.	Important	http://www.loa-cnr.it/ontologies/DUL.owl#describes http://www.demcare.eu/ontologies/contextdescriptor.owl#describes http://www.demcare.eu/ontologies/contextdescriptor.owl#isDescribedBy http://www.loa-cnr.it/ontologies/DUL.owl#isDescribedBy http://xmlns.com/foaf/0.1/givenname http://xmlns.com/foaf/0.1/givenName http://xmlns.com/foaf/0.1/family_name http://xmlns.com/foaf/0.1/familyName
P13	Missing inverse relationships	This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".	Minor	http://www.demcare.eu/ontologies/contextdescriptor.owl#dependency http://xmlns.com/foaf/0.1/accountServiceHomepage http://xmlns.com/foaf/0.1/openid **+25 more**
P20	Misusing ontology	The contents of some annotation properties are swapped or misused. An example of this type of pitfall is to	Minor	http://www.loa-cnr.it/ontologies/DUL.owl#isAg

	annotations	include in the Label annotation of the class "Crossroads" the following sentence the place of intersection of two or more roads; and to include in the Comment annotation the word 'Crossroads'.		entInvolvedIn
P22	Using different naming criteria in the ontology	Ontology elements are not named using the same convention within the whole ontology. It is considered a good practice that the rules and style of lexical encoding for naming the different ontology elements is homogeneous within the ontology. One possibility for rules is that concept names start with capital letters and property names start with non-capital letters. In the case of style, there are different options such as camel case, hyphen style, underscore style, and the combinations.		http://xmlns.com/foaf/0.1/topic_interest http://xmlns.com/foaf/0.1/workplaceHomepage
P24	Using recursive definition	An ontology element is used in its own definition. For example, it is used to create the relationship "hasFork" and to establish as its range the following the set of restaurants that have at least one value for the relationship "hasFork".		http://www.loa-cnr.it/ontologies/DUL.owl#Task http://www.loa-cnr.it/ontologies/DUL.owl#Role http://www.loa-cnr.it/ontologies/DUL.owl#SocialObject **+13 more**
P25	Defining a relationship inverse to itself	A relationship is defined as inverse of itself. In this case, this property could have been defined as "owl:SymmetricProperty" instead.	Important	http://www.loa-cnr.it/ontologies/DUL.owl#isRelatedToConcept http://www.loa-cnr.it/ontologies/DUL.owl#isRelatedToDescription http://www.loa-cnr.it/ontologies/DUL.owl#nearTo http://www.loa-cnr.it/ontologies/DUL.owl#hasCommonBoundary
P26	Defining inverse relationships for a symmetric one	A relationship is defined as "owl:SymmetricProperty" and there is also a relationship (it could be itself or another relationship) defined as its inverse.	Important	http://www.loa-cnr.it/ontologies/DUL.owl#nearTo http://www.loa-cnr.it/ontologies/DUL.owl#hasCommonBoundary http://www.loa-cnr.it/ontologies/DUL.owl#isRelatedToDescription http://www.loa-

				cnr.it/ontologies/DUL.owl#isRelatedToConcept
P30	Missing equivalent classes	When an ontology is imported into another, classes with the same conceptual meaning that are duplicated in both ontologies should be defined as equivalent classes to benefit the interoperability between both ontologies. However, the ontology developer misses the definition of equivalent classes in the cases of duplicated concepts. An example of this pitfall can be not to have the equivalent knowledge explicitly defined between "Trainer" (class in the imported ontology) and "Coach" (class in the ontology about sports being developed).	Important	http://www.loa-cnr.it/ontologies/DUL.owl#Task http://xmlns.com/foaf/0.1/Project http://www.loa-cnr.it/ontologies/DUL.owl#Project http://www.loa-cnr.it/ontologies/DUL.owl#Situation http://www.loa-cnr.it/ontologies/DUL.owl#Place http://www.loa-cnr.it/ontologies/DUL.owl#Plan http://www.loa-cnr.it/ontologies/DUL.owl#Design http://www.loa-cnr.it/ontologies/DUL.owl#Pattern
P31	Defining wrong equivalent classes	Two classes are defined as equivalent when they are not necessarily equivalent. For example, defining Car as equivalent to Vehicle	Critical	http://xmlns.com/foaf/0.1/Image http://schema.org/ImageObject http://schema.org/CreativeWork http://xmlns.com/foaf/0.1/Document
P34	Untyped class	A resource is used as a class, e.g. appearing as the object of an "rdf:type", "rdfs:domain", or "rdfs:range" statement, or as the subject or object of an "rdfs:subClassOf" statement, without having been declared as a Class.	Important	http://schema.org/ImageObject http://purl.org/dc/terms/Agent http://schema.org/CreativeWork http://schema.org/Person http://www.w3.org/2000/10/swap/pim/contact#Person
P35	Untyped property	A resource is used as a property, e.g. appearing as the subject or object of an "rdfs:subPropertyOf" statement, without having been declared as a "rdf:Property" or some subclass of it.	Important	http://purl.org/dc/terms/creator
P36	URI contains file extension	Guidelines suggest avoiding file extension in persistent URIs, particularly those related to the technology used, as for example ".php" or ".py". In our case we have adapted it to the ontology web languages used to	Minor	http://www.demcare.eu/ontologies/contextdescriptor.owl

		formalized ontologies and their serializations. In this regard, we consider as pitfall including file extensions as ".owl", ".rdf", ".ttl", ".n3" and ".rdfxml" in an ontology URI. An example of this pitfall (at 29th June, 2012) could be found in the "BioPAX Level 3 ontology (biopax)" ontologyOs URI (http://www.biopax.org/release/biopax-level3.owl) that contains the extension ".owl" related to the technology used.		
P40	Namespace hijacking	This means reusing or referring to terms from other namespaces not actually defined in such namespace. This pitfall is related to the Linked Data publishing guidelines provided in [6], "Only define new terms in a namespace that you control". Example: the "WSMO-Lite Ontology (wl)" which URI is http://www.wsmo.org/ns/wsmo-lite#, uses http://www.w3.org/2000/01/rdf-schema#Property that is not defined in the rdf namespace (http://www.w3.org/2000/01/rdf-schema#) instead of using http://www.w3.org/1999/02/22-rdf-syntax-ns#Property, that is ac-tually defined in the rdfs namespace (http://www.w3.org/1999/02/22-rdf-syntax-ns#).	Critical	http://creativecommons.org/ns#license

Table 14 Various ontology metrics calculated for the Context Descriptor ontology

DL expressivity	SRIQ(D)
Axioms	1814
Logical axiom count	691
Number of Classes	186
Number of Internal Classes	0
Number of Named Classes	92
Number of Anonymous Classes	94
Number of Restrictions	81
Number of Complement Classes	0
Number of Enumerated Classes	0
Number of Intersection Classes	6
Number of Union Classes	7
Number of Leaf Classes	80

Number of Root Classes	102
Maximum Depth of Class Inheritance Tree	6
Average Depth of Class Inheritance Tree	2.95
Maximum Number of Class Ancestors	11
Average Number of Class Ancestors	3.41
Number of Properties	184
Number of Internal Properties	0
Number of Object Properties	135
Number of Functional Properties	1
Number of Inverse Functional Properties	5
Number of Symmetric Properties	4
Number of Inverse Properties	106
Number of Transitive Properties	4
Number of Datatype Properties	34
Number of Annotation Properties	9
Number of Individuals	13
Number of Internal Individuals	0
Number of Named Individuals	13
Number of Anonymous Individuals	0

6.1.3 Home/NHome Ontology

Regarding the ontology that contains the knowledge constructs for the home domain (activities, problems, etc.), OOPS! identified 7 pitfalls. As it is depicted in Table 15, the majority of the pitfalls refers to missing annotations and other minor problems that we have successfully addressed. It should be noted that the Home/NHome ontology also contains the knowledge structures needed to model questionnaires. Table 16 depicts various ontology metrics as these have been calculated by Protégé and OntoMetricCalc Java API.

Table 15 OOPS! evaluation results for the Home/NHome ontology

Code	Name	Description	Importance	#Affected elements
P04	Creating unconnected ontology elements	Ontology elements (classes, relationships or attributes) are created with no relation to the rest of the ontology. An example of this type of pitfall is to create the relationship "memberOfTeam" and to miss the class representing teams; thus, the relationship created is isolated in the ontology.	Minor	http://www.demcare.eu/ontologies/event.owl#Person
P08	Missing annotations	Ontology terms lack annotations properties. This kind of properties improves the ontology understanding	Minor	http://www.demcare.eu/ontologies/home.owl#PeriodicLimbMov

		and usability from a user point of view.		ement **+ 99 more**
P11	Missing domain or range in properties	Relationships and/or attributes without domain or range (or none of them) are included in the ontology. There are situations in which the relation is very general and the range should be the most general concept "Thing". However, in other cases, the relations are more specific and it could be a good practice to specify its domain and/or range. An example of this type of pitfall is to create the relationship "hasWritten" in an ontology about art in which the relationship domain should be "Writer" and the relationship range should be "LiteraryWork".	Important	http://www.demcare.eu/ontologies/home.owl#stressLevel http://www.demcare.eu/ontologies/home.owl#eatingLocation **+34 more**
P13	Missing inverse relationships	This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".	Minor	http://www.w3.org/2006/time#hasDurationDescription http://www.demcare.eu/ontologies/home.owl#hasQuestion **+ 9 more**
P22	Using different naming criteria in the ontology	Ontology elements are not named using the same convention within the whole ontology. It is considered a good practice that the rules and style of lexical encoding for naming the different ontology elements is homogeneous within the ontology. One possibility for rules is that concept names start with capital letters and property names start with non-capital letters. In the case of style, there are different options such as camel case, hyphen style, underscore style, and the combinations. Some notions about naming conventions are provided in [2].	Minor	http://www.demcare.eu/ontologies/home.owl#q_score http://www.demcare.eu/ontologies/home.owl#numberOfAwakenings
P34	Untyped class	A resource is used as a class, e.g. appearing as the object of an "rdf:type", "rdfs:domain", or "rdfs:range" statement, or as the subject or object of an "rdfs:subClassOf" statement, without having been declared as a Class.	Important	**http://www.demcare.eu/ontologies/descriptive.owl#hasReportingTime**
P36	URI contains file extension	Guidelines suggest avoiding file extension in persistent URIs, particularly those related to the technology used, as for example ".php" or ".py". In our case we have adapted it	Minor	http://www.demcare.eu/ontologies/home.owl

		to the ontology web languages used to formalized ontologies and their serializations. In this regard, we consider as pitfall including file extensions as ".owl", ".rdf", ".ttl", ".n3" and ".rdfxml" in an ontology URI. An example of this pitfall (at 29th June, 2012) could be found in the "BioPAX Level 3 ontology (biopax)" ontologyOs URI (http://www.biopax.org/release/biopax-level3.owl) that contains the extension ".owl" related to the technology used.		

Table 16 Various ontology metrics calculated for the Home/NHome ontology

Axioms	880
Logical axiom count	498
DL expressivity	ALCHOI
Number of Classes	125
Number of Internal Classes	0
Number of Named Classes	51
Number of Anonymous Classes	74
Number of Restrictions	74
Number of Complement Classes	0
Number of Enumerated Classes	0
Number of Intersection Classes	0
Number of Union Classes	0
Number of Leaf Classes	37
Number of Root Classes	76
Maximum Depth of Class Inheritance Tree	3
Average Depth of Class Inheritance Tree	1.47
Maximum Number of Class Ancestors	3
Average Number of Class Ancestors	1.47
Number of Properties	49
Number of Internal Properties	0
Number of Object Properties	11
Number of Functional Properties	0
Number of Inverse Functional Properties	0
Number of Symmetric Properties	0
Number of Inverse Properties	0
Number of Transitive Properties	0

Number of Datatype Properties	38
Number of Annotation Properties	0
Number of Individuals	0
Number of Internal Individuals	0
Number of Named Individuals	0
Number of Anonymous Individuals	0

7 Integration of Components and Usage in Pilots

This section aims to give an overview of the WP5 integrated modules and their usage across pilots, interlinking the progress made in the present work package with integration and clinical piloting (WP7 and WP8). This overview concerns not only the most recent developments, presented in this deliverable, but in the entire work package, effectively reflecting its total contribution to clinical dementia care. It also lists the most notable results, while pointing the reader to respective deliverables for further reading.

Table 17 captures all methods produced in WP5, the components in which they were integrated and their usage in pilots across the consortium. Further details are given below according to each integrated component.

Complex Activity Recognition - CAR is a processing component, based in visual data input in the form of image and depth information (RGB-D). The way the component extracts visual characteristics in real-time, from these data is detailed throughout WP5. However, CAR also integrates further analysis of these primary results by means of further models and rules for the extraction of higher-level characteristics. In detail, the methods employed in CAR enable location detection according to predefined zones in a scene, GAIT attributes, such as speed and stride length. Rule constructs are triggered in real-time to infer the detection of events and Activities of Daily Living (ADLs) e.g. according to spatial and temporal criteria (staying within a zone for a period of time). The methods combined to obtain this functionality entail unsupervised event models and semantic event fusion and are fully listed on Table 17. Regarding piloting, the component is responsible for all event fusion and result extraction in @Lab Nice. It was also used throughout most pilots for location and ADL detection in the entire WP8 results. In @Home Thessaloniki and Dublin it is substituted by other visual components (HAR and WCPU respectively) and SI.

The *Wearable Camera Processing Unit – WCPU* is quite similarly a visual-based processing component, which further aggregates initial visual output, as described in the entire WP5 work, into higher-level concepts. The component specializes in detection or rooms, objects in usage and Activities of Daily Living (*Room Recognition*, *Object Recognition* and *Activity Recognition from Wearable Camera – ORWC*, *RRWC*, *ARWC*). In this work package, visual recognition methods were infused with supervised learning, patient-tailored models, sensor fusion and aggregation techniques, as listed on Table 17. The component requires a fair amount of work to collect and time to process data, due to its complexity and was therefore used only in @Home Dublin (D8.5) and in certain data collection experiments of @Lab Nice.

The *Semantic Interpretation – SI* processing component is a major outcome of WP5, providing sensor-independent fusion and detection of problems, correlations and ADLs. The module incorporates reasoning capabilities using the Dem@Care ontologies to fuse together events coming from any type of sensors, whether it is raw visual or lifestyle sensor information or even existing ADLs from other components. SI combines all this information to extract a final, high-level activity using rules. It can also extract clinical problems (based on rules), correlations, trends and extract statistics. The full list of methods is presented in Table 17. SI was used extensively in piloting, providing statistics in @Lab Thessaloniki, problems, patterns and correlations in @Home Dublin and Thessaloniki and @NH Luleå. ADL recognition was heavily used in @Home Thessaloniki and for further WCPU aggregation in @Home Dublin.

Knowledge Base Manager – KBM is the Dem@Care component used to assert knowledge into the system. It incorporates the Dem@Care ontologies, an RDF triple store and certain APIs integrated in the system to universally store information and enable their further interpretation by SI. The component is naturally an essential part at all pilots, regardless of sensor deployments, entailing the methods listed on Table 17.

Table 17 Integration of all WP5 components and usage in pilots

Method	**Modalities**	**Integration**	**Usage in Pilots**				
			@Lab		**@NH**	**@Home**	
			Nice	**Thess**	**Luleå**	**Dublin**	**Thess**
CAR							
Hierarchical Model-based approach for Activity Recognition (D5.2)	Location, GAIT, ADLs	✓	✓	✓	✓	(✓)	(✓)
Unsupervised Activity Recognition Using Fixed Cameras (D5.3)							
Uncertainty Modelling for Low-level Event Detection (D5.4)							
Enhancing Pre-defined Event Models Using Unsupervised Event Models (D5.5)							
Semantic Event Fusion of Different Visual Modality Concepts (D5.6)							
WCPU							
Supervised, Patient-tailored Activity Pattern Recognition in Wearable Video (D5.2, D5.3, D5.5)	Location, Objects, ADLs	✓	(✓)	-	-	✓	-
Probabilistic Activity Recognition and Confidence Values (D5.4)							
Event Recognition from Wearable Video Cameras and Sensor Fusion (D5.6)							

SI							
Rule-based Activity Interpretation and Fusion (D5.1, D5.2)	Problems, Patterns, Statistics, Correlations, ADLs	✓	-	✓	✓	✓	✓
Context-based Fusion in Multi-sensor Environments (D5.4)							
Ontology Patterns for Behaviour Modelling (D5.3, D5.5)							
Situation Descriptors, Context Connections and Rules (D5.6)							
KBM							
Semantic Knowledge Structures and Representation (D5.1)	-	✓	✓	✓	✓	✓	✓
Semantic Repository and Interfaces (D5.1, D5.2)							

8 Conclusions

This deliverable presented the third and final version of the multi-parametric behaviour interpretation framework. More specifically, we described an XML-based format for representing events, which provides a more human-readable syntax than the one used in previous versions of the framework. We also presented the execution time of activity recognition modules from wearable camera. Despite the challenging activity recognition task, our framework is able to process one frame per 2 seconds. Finally, we investigated improvements on the activity recognition performance by adding movement tracking modality to the video/image analysis. Future directions in this research include the further improvement of activity recognition scalability toward (near) real-time performance and the incorporation of additional modalities for improving the accuracy of activity recognition.

We have also presented a multimedia event recognition framework to semantically align different, multimodal visual sensors. The framework allows the modelling and incorporation of scene semantics, the hierarchical nature of high-level events, using a knowledge-driven approach, and the fusion of the multiple sources of conceptual data using a semi-probabilistic approach based on the explicit modelling of concept relevance and sensor reliability into the event inference step. We demonstrated that the proposed framework performs a more accurate fusion on the presence of partial information than a fully supervised approach (SVM-based), and it is capable of delimiting the temporal boundaries of activities intervals more accurately than an ontology-driven approach that reasons with a holistic view of the complete set of observed concepts in the multimedia recording. Finally, the proposed method is also more robust to incomplete and unreliable evidence than the baseline approaches. By employing a novel semantic-based method to synchronize sensor concept streams in multimedia scenarios, we also leverage the event recognition performance of the framework and its individual visual concept detectors without provoking decay in the performance of naturally brief events.

The combination of heterogeneous visual modalities for activity recognition also proved to be successful approach, since the framework can outperform the individual sensors event recognition most of the time, even in the absence of information from the complete set of sensors. Future work will investigate ways to improve the semantic-based synchronization on dense concept streams and to adapt it for real-time scenarios, and to dynamically update the estimations of concept reliability for concept fusion in response to observed changes on scene characteristics or on the quality of visual concept detectors.

Finally, we described the implementation of the RDF-based activity recognition framework using SPARQL rules and we have elaborated on the Dem@Care core ontologies, presenting evaluation results and relevant ontology metrics. The framework detects complex and interleaved activities based on loosely coupled domain activity dependencies rather than on strict contextual constraints. As future directions of our research, we plan to extend the framework by incorporating habitual knowledge for defining the domain context descriptors of the framework. We also plan to use the framework on top of CAR, so as to further investigate improvements on the multi-fusion activity recognition framework in WP5.

9 References

[1] J.F. Allen, “Maintaining Knowledge About Temporal Intervals,” Commun. ACM, vol. 26, no. 11, pp. 832–843, Nov. 1983.

[2] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, “A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation,” Pattern Anal. Mach. Intell. IEEE Trans., vol. 31, no. 9, pp. 1685–1699, Sep. 2009.

[3] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris, “Activity Detection using Sequential Statistical Boundary Detection (SSBD),” under Rev. to Comput. Vis. Image Underst., 2015.

[4] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris, “Recognition of Activities of Daily Living for Smart Home Environments,” in Intelligent Environments (IE), 2013 9th International Conference on, 2013, pp. 173–180.

[5] F. Baader and U. Sattler, “An Overview of Tableau Algorithms for Description Logics,” Stud. Log., vol. 69, no. 1, pp. 5–40, 2001.

[6] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, Jun. 2008.

[8] A.B. Benevides, G. Guizzardi, “A Model-Based Tool for Conceptual Modeling and Domain Ontology Engineering in OntoUML”, 11th ICEIS, Milan (2009).

[9] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Sci. Am., vol. 284, no. 5, pp. 34–43, May 2001.

[10] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” Pervasive Mob. Comput., vol. 6, no. 2, pp. 161–180, 2010.

[11] H. Boujut, J. Benois-Pineau, and R. Megret, “Fusion of multiple visual cues for visual saliency extraction from wearable camera settings with strong motion,” in ECCV 2012 - Workshops, 2012, pp. 436–445.

[12] Y. Cao, L. Tao, and G. Xu, “An event-driven context model in elderly health monitoring,” in Proceedings of Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2009.

[13] D.P. Chau, F. Bremond, and M. Thonnat, “A multi-feature tracking algorithm enabling adaptation to context.” 2011.

[14] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-Based Activity Recognition,” Syst. Man, Cybern. Part C Appl. Rev. IEEE Trans., vol. 42, no. 6, pp. 790–808, Nov. 2012.

[15] L. Chen, C. Nugent, and G. Okeyo, “An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes,” Human-Machine Syst. IEEE Trans., vol. 44, no. 1, pp. 92–105, Feb. 2014.

[16] L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” Knowl. Data Eng. IEEE Trans., vol. 24, no. 6, pp. 961–974, 2012.

[17] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[18] C.F. Crispim-Junior, V. Bathrinarayanan, B. Fosty, A. Konig, R. Romdhane, M. Thonnat, and F. Bremond, “Evaluation of a monitoring system for event recognition of older people,” in Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on, 2013, pp. 165–170.

[19] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in In Workshop on Statistical Learning in Computer Vision, ECCV, 2004, pp. 1–22.

[20] S. Dasiopoulou, V. Efstathiou, G. Meditskos. D5.1 Semantic Knowledge Structures and Representation, Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support, Dem@Care – FP7 288199.

[21] S. Dasiopoulou, V. Efstathiou, G. Meditskos, C. Crispim-Junior, A.T. Nghiem, V. Buso, “D5.2 Multi-parametric Behaviour Interpretation v1”, Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support, Dem@Care – FP7 288199

[22] V. Dovgalecs, R. Mégret, and Y. Berthoumieu, “Multiple Feature Fusion Based on Co-Training Approach and Time Regularization for Place Classification in Wearable Video,” Adv. Multimed., 2013.

[23] V.P. Dragalin, “Optimality of generalized Cusum procedure in quickest detection problem,” Proc. Steklov Inst. Math. Transl., vol. 202, pp. 107–120, 1994.

[24] G. Farnebäck, “Fast and Accurate Motion Estimation using Orientation Tensors and Parametric Motion Models,” in Proceedings of 15th International Conference on Pattern Recognition, 2000, vol. 1, pp. 135–139.

[25] A. Fleury, N. Noury, and M. Vacher, “Introducing knowledge in the process of supervised classification of activities of Daily Living in Health Smart Homes,” in Proceedings of 12th IEEE International Conference on e-Health Networking Applications and Services, 2010, pp. 322–329.

[26] M.F. Folstein, L. N. Robins, and J. E. Helzer, “THe mini-mental state examination,” Arch. Gen. Psychiatry, vol. 40, no. 7, p. 812, 1983.

[27] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, “Early Versus Late Fusion in Semantic Video Analysis,” in Proceedings of the 13th Annual ACM International Conference on Multimedia, 2005, pp. 399–402.

[28] L. Gao, A. K. Bourke, and J. Nelson, “A system for activity recognition using multi-sensor fusion,” in Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, 2011, pp. 7869–7872.

[29] A. Gómez-Pérez, "Ontology Evaluation", Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems, pp. 251-274, 2004.

[30] I. González Díaz, V. Buso, J. Benois-Pineau, G. Bourmaud, and R. Megret, "Modeling Instrumental Activities of Daily Living in Egocentric Vision As Sequences of Active Objects and Context for Alzheimer Disease Research," in Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare, 2013, pp. 11–14.

[31] B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler., "{OWL} 2: The Next Step for {OWL}," Web Semant. Sci. Serv. Agents World Wide Web, vol. 6, no. 4, pp. 309–322, Oct. 2008.

[32] G. Guizzardi, "Ontological foundations for structural conceptual models", Centre for Telematics and Information Technology, University of Twente, The Netherlands, (2005).

[33] D.M. J. Tax and R. P. W. Duin, "Support vector data description," Mach. Learn., vol. 54, no. 1, pp. 45–66, 2004.

[34] I.H. Jhuo, G. Ye, S. Gao, D. Liu, Y.-G. Jiang, D. T. Lee, and S.-F. Chang, "Discovering Joint Audio-Visual Codewords for Video Event Detection," Mach. Vis. Appl., vol. 25, no. 1, pp. 33–47, 2014.

[35] S. Karaman, J. Benois-Pineau, R. Mégret, V. Dovgalecs, J.-F. Dartigues, and Y. Gaëstel, "Human Daily Activities Indexing in Videos from Wearable Cameras for Monitoring of Patients with Dementia Diseases," in International Conference on Pattern Recognition (ICPR), 2010, 2010, pp. 4113–4116.

[36] K.M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity Forecasting," in ECCV (4)'12, 2012, pp. 201–214.

[37] A. Klaser, M. Marszalek, C. Schmid, and A. Zisserman, "Human Focused Action Localization in Video," Proc. 11th Eur. Conf. Trends Top. Comput. Vis. - Vol. Part I, pp. 219–233, 2012.

[38] N.C. Krishnan and D. J. Cook, "Activity Recognition on Streaming Sensor Data," Pervasive Mob. Comput., vol. 10, pp. 138–154, Feb. 2014.

[39] M. Land, N. Mennie, and J. Rusted, "The roles of vision and eye movements in the control of activities of daily living.," Perception, vol. 28, pp. 1311–1328, 1999.

[40] G. Lavee, E. Rivlin, and M. Rudzsky, "Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video," Syst. Man, Cybern. Part C Appl. Rev. IEEE Trans., vol. 39, no. 5, pp. 489–504, Sep. 2009.

[41] Q.V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3361–3368.

[42] G. Luo, S. Yang, G. Tian, C. Yuan, W. Hu, and S. J. Maybank, "Learning Human Actions by Combining Global Dynamics and Local Appearance," Pattern Anal. Mach. Intell. IEEE Trans., vol. 36, no. 12, pp. 2466–2482, Dec. 2014.

[43] M. Marszalek and C. Schmid, "Spatial Weighting for Bag-of-Features," in IEEE Conference on Computer Vision & Pattern Recognition, 2006, vol. 2, pp. 2118–2125.

[44] G. Meditskos, E. Kontopoulos, and I. Kompatsiaris, "Knowledge-Driven Activity Recognition and Segmentation Using Context Connections," in 13th International Semantic Web Conference (ISWC'14), 2014, pp. 260–275.

[45] G. Meditskos, E. Kontopoulos, C. Crispim-Junior, S. Cosar, F. Bremond, R. Mégret, G. Usseglio, Y. Berthoumieu, V. Buso, J. Benois-Pineau, D. Stampouli, "D5.4 Multi-parametric Behaviour Interpretation v2", Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support, Dem@Care – FP7 288199

[46] H. Medjahed, D. Istrate, J. Boudy, J. L. Baldinger, and B. Dorizzi, "A pervasive multi-sensor data fusion for smart home healthcare monitoring," in Proceedings of IEEE International Conference on Fuzzy Systems, 2011, pp. 1466–1473.

[47] G.K. Myers, R. Nallapati, J. van Hout, S. Pancoast, R. Nevatia, C. Sun, A. Habibian, D. C. Koelma, K. E. A. van de Sande, A. W. M. Smeulders, and C. G. M. Snoek, "Evaluating Multimedia Features and Fusion for Example-based Event Detection," Mach. Vis. Appl., vol. 25, no. 1, pp. 17–32, Jan. 2014.

[48] R. Nevatia, J. Hobbs, and B. Bolles, "An Ontology for Video Event Representation," in Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 7, 2004, pp. 119–129.

[49] A.T. Nghiem and F. Bremond, "Background subtraction in people detection framework for RGB-D cameras," in Proceedings of 11th IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2014.

[50] D. Nute, "Defeasible Reasoning*", Proc. 20th Int. Conference on Systems Science*, IEEE Press, pp. 470-477, 1987.

[51] S. Oh, S. McCloskey, I. Kim, A. Vahdat, K. J. Cannons, H. Hajimirsadeghi, G. Mori, A. G. A. Perera, M. Pandey, and J. J. Corso, "Multimedia Event Detection with Multimodal Feature Fusion and Temporal Concept Localization," Mach. Vis. Appl., vol. 25, no. 1, pp. 49–69, Jan. 2014.

[52] J. Pinquier, S. Karaman, L. Letoupin, P. Guyot, R. Mégret, J. Benois-Pineau, Y. Gaëstel, and J.-F. Dartigues, "Strategies for multiple feature fusion with Hierarchical HMM: Application to activity recognition from wearable audiovisual sensors.," in International Conference on Pattern Recognition (ICPR), 2012, pp. 3192–3195.

[53] H. Pirsiavash and D. Ramanan, "Detecting Activities of Daily Living in First-person Camera Views," in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[54] J.C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in Advances in Large Margin Classifiers, 1999, pp. 61–74.

[55] M. Poveda, M.C. Suárez-Figueroa, A. Gómez-Pérez, "A Double Classification of Common Pitfalls in Ontologies", OntoQual 2010 - Workshop on Ontology Quality at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010). Proceedings of the Workshop on Ontology Quality -

OntoQual 2010. ISBN: ISSN 1613-0073. CEUR Workshop Proceedings, pp. 1-12. 15 October 2010, Lisbon, Portugal

[56] M. Poveda-Villalón, A., Gómez-Pérez, M.C., Suárez-Figueroa, "Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation, International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), 7-34, 2014.

[57] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal Semantic Place Classification," Int. J. Robot. Res. (IJRR), Spec. Issue Robot. Vis., vol. 29, no. 2–3, pp. 298–320, Feb. 2010.

[58] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, C. Wroe, "Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns", In Proc. of EKAW 2004, pp: 63–81. Springer. 2004.

[59] L. Rong and L. Ming, "Recognizing Human Activities Based on Multi-Sensors Fusion," in Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on, 2010, pp. 1–4.

[60] S. Salvador and P. Chan, "Toward Accurate Dynamic Time Warping in Linear Time and Space," Intell. Data Anal., vol. 11, no. 5, pp. 561–580, Oct. 2007.

[61] J.C. SanMiguel and J. M. Martínez, "A semantic-based probabilistic approach for real-time video event recognition," Comput. Vis. Image Underst., vol. 116, no. 9, pp. 937–952, 2012.

[62] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem." 1996.

[63] R. Shaw, R. Troncy, L. Hardman, "LODE: Linking Open Descriptions of Events. In Proceedings of the 4th Asian Conference on The Semantic Web (ASWC '09), Asunción Gómez-Pérez, Yong Yu, and Ying Ding (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 153-167, 2009.

[64] V. Sreekanth, A. Vedaldi, C. V Jawahar, and A. Zisserman, "Generalized {RBF} feature maps for efficient detection," in Proceedings of the British Machine Vision Conference (BMVC), 2010.

[65] G. Stevenson, S. Knox, S. Dobson, and P. Nixon, "Ontonym: a collection of upper ontologies for developing pervasive systems," in Proceedings of the 1st Workshop on Context, Information and Ontologies, 2009, pp. 9:1–9:8.

[66] G. Stevenson, J. Ye, S. Dobson, and P. Nixon, "LOC8: A location Model and Extensible Framework for Programming with location," IEEE Pervasive Comput., vol. 9, no. 1, pp. 28–37, Jan. 2010.

[67] T. Vu, F. Bremond, and M. Thonnat, "Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition," in The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), 2003.

[68] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3169–3176.

[69] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, “Dense trajectories and motion boundary descriptors for action recognition,” Int. J. Comput. Vis., vol. 103, no. 1, pp. 60–79, 2013.

[70] J. Ye, L. Coyle, S. Dobson, and P. Nixon, “Ontology-based Models in Pervasive Computing Systems,” Knowl. Eng. Rev., vol. 22, no. 04, pp. 315–347, Dec. 2007.

[71] N. Zouba, F. Bremond, and M. Thonnat, “An Activity Monitoring System for Real Elderly at Home: Validation Study,” in 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2010, 2010.