



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

*Corso di Laurea in Informatica per la Comunicazione
Digitale*

VALUTAZIONE
DELL'ACCETTABILITÀ DI SISTEMI DI
RICONOSCIMENTO DI ATTIVITÀ
UMANE BASATI SU ACTIVE
LEARNING

Relatore: Dott. Gabriele CIVITARESE

Correlatore: Dott. Riccardo PRESOTTO, Prof. Claudio BETTINI

Tesi di:
Emanuela ELLI
Matricola: 931039

Anno Accademico 2020-2021

Ringraziamenti

A conclusione di questo lavoro, desidero ringraziare il Professor Bettini che mi ha dato la possibilità di iniziare questo tirocinio ed un altrettanto ringraziamento lo dedico a Gabriele e Riccardo per avermi sostenuto e guidato durante lo svolgimento di tutto il lavoro portandomi al termine di questo percorso.

Devo dei ringraziamenti speciali ai miei genitori, Elena ed Emanuele, che mi hanno spronato ad intraprendere questo percorso, cercando di valorizzare le mie capacità, senza mai andare contro i miei desideri. Grazie anche a mia sorella Eleonora che mi ha sopportato durante le mille difficoltà incontrate nello studio cercando di sdrammatizzare quei momenti.

Un altrettanto importante ringraziamento lo dedico al mio ragazzo, Luca. Oltre ad avermi sopportato ogni giorno durante questi tre anni, tra sofferenze e gioie, mi ha sostenuto e confortato in ogni momento credendo sempre nelle mie potenzialità, anche più di me stessa, cercando di tirare fuori sempre il meglio di me.

All'interno della vita universitaria un ringraziamento particolare lo dedico a Martina. Nonostante la pandemia, nonostante la scelta di corsi differenti e nonostante la lontananza, ci siamo sempre aiutate, confrontate e supportate a vicenda. Grazie per essermi stata accanto in questi anni ed aver affrontato questo percorso insieme.

Grazie a tutti quei compagni o compagne che, anche per brevi tratti, hanno reso questi momenti di studio e di lavoro più sereni e leggeri tra una chiaccherata e una risata, con l'augurio di condividere assieme altri momenti così.

Indice

Ringraziamenti	ii
Introduzione	1
1 Stato dell'arte	8
1.1 Riconoscimento di attività umane	9
1.2 Caratterizzazione delle attività	12
1.3 Tecniche di active learning e il loro utilizzo	14
1.4 Problemi riguardo l'usabilità di sistemi basati su active learning	15
1.5 Wizard of Oz	16
2 Metodologia sperimentale proposta	21
2.1 Obbiettivi dell'esperimento	23
2.2 Descrizione dell'esperimento basato su WOz	26
2.3 Caratterizzazione del contesto	27
2.4 Ciclo di vita di una query	29
2.5 Definizione delle attività	32
2.6 Modalità sperimentale	33
2.7 Invio della query	35
2.8 Definizione della tipologia di prompt e delle interfacce	37
2.9 Valutazione post-esperimento	38
3 Caso d'uso: riconoscimento di attività in smart-home lab	40
3.1 Requisiti del sistema	41
3.2 Setup sperimentale	45
3.3 Fasi di esecuzione dell'esperimento	46
3.3.1 Definizione scenari ed attività	48
4 Implementazione di un prototipo	50
4.1 Architettura	51
4.2 Interfacce	53

4.2.1	Interfaccia wizard	53
4.2.2	Interfaccia tablet	64
4.3	Memorizzazione dei dati	66
4.4	Comunicazione tramite socket	69
5	Conclusione e sviluppi futuri	71
	Bibliografia	73

Introduzione

Il riconoscimento dell'attività umana, o in inglese *human activity recognition* (HAR), è un ampio campo di studio che si occupa di identificare il movimento o l'azione specifica di un individuo in base ai dati ricavati dai sensori. I movimenti sono spesso attività quotidiane, come camminare, parlare, stare in piedi e sedersi, ma possono anche essere attività più mirate quali attività svolte in cucina oppure attività svolte all'interno di una fabbrica. I dati del sensore possono essere registrati in remoto ad esempio in ambienti *smart* (quali le *smart-home*), come video, radar o altri metodi wireless. In alternativa, i dati possono essere registrati direttamente sul soggetto, ad esempio indossando dispositivi dotati di accelerometri, giroscopi, ricevitori GPS, cardiofrequenzimetri o altre tipologie di sensori. L'utilizzo di tali strumenti abilitano il tracciamento continuo delle attività quotidiane svolte dagli individui e pertanto ne permettono il riconoscimento.

Negli ultimi decenni è stato dedicato un grande sforzo, sia nella ricerca che nell'industria, al riconoscimento dell'attività umana basato su sensori e in particolare un ampio settore di ricerca si è focalizzato su sensori non intrusivi in parte per evitare problemi di privacy date da altre soluzioni. Tale ambito di ricerca è utilizzato per diverse finalità, quali:

- fornire assistenza ad anziani, ad esempio i *caregiver* utilizzano sensori per tracciare ed analizzare le attività della vita quotidiana (ADL) delle persone anziane al fine di fornire loro un'assistenza proattiva [1];
- lo sviluppo di applicazioni di *healthcare* (ovvero applicazioni volte all'assistenza sanitaria), ad esempio per monitorare problemi comportamentali come sintomi precoci di degrado cognitivo;
- lo sviluppo di applicazioni *wellbeing* (il “digital wellbeing”, in italiano “benessere digitale”, è uno stato di buona salute psicofisica derivante da un rapporto sano con le tecnologie digitali che si ottiene tramite un equilibrio tra attività online e vita reale, creando consapevolezza per un corretto utilizzo degli strumenti informatici e per il controllo dei loro effetti), ad esempio la raccomandazione

musicale sensibile al contesto [2] affinché il sistema sia in grado di consigliare un tipo specifico di musica adatto all'attività che l'utente sta svolgendo;

- la realizzazione di ambienti intelligenti ottimizzati sul risparmio energetico in base alle attività svolte e le routine delle persone che vi abitano.

Raggiungere tali obiettivi però non risulta essere un compito facile. Il motivo principale che rende il riconoscimento di attività umane basato su sensori un ambito complesso è che l'interpretazione dei dati provenienti dai sensori richiede tecniche statistiche (siccome non è possibile utilizzare tecniche analitiche). Inoltre è possibile incombere in situazioni in cui diversi individui svolgono le stesse attività con modalità differenti oppure svolgono attività diverse in maniera simile. Il primo caso è motivato dal fatto che le modalità con cui si svolgono le attività possono essere molto diverse in base alle caratteristiche fisiche ed all'età dei soggetti. Il secondo caso, invece, è motivato dal fatto che molte attività hanno pattern di movimento simili e pertanto risulta difficile distinguerle solo tramite l'utilizzo di sensori. Ad aggiungere un ulteriore grado di complessità è anche la possibilità che i sensori generino dei dati "rumorosi" ovvero errati o anomali. Tali considerazioni inducono ambiguità nell'interpretazione delle attività. Pertanto, siccome il mondo reale è colmo di diverse tipologie di incertezze, per raggiungere questo obiettivo è necessario stabilire dei metodi per imparare dai dati, estrarre conoscenza ed aiutare ad ottenere dei risultati più precisi dalla classificazione.

Data la complessità nello svolgimento di attività, solitamente nella ricerca sul riconoscimento di attività umane vengono utilizzati contemporaneamente diversi sensori per cercare di catturare più aspetti possibili dai movimenti effettuati dal soggetto. Dopo l'estrazione, i dati raccolti dai sensori devono essere pre-processati e passati come input ad algoritmi di *machine learning*. Il modo in cui i dati dei sensori di basso livello vengono mappati su astrazioni di livello superiore è la chiave per il riconoscimento dell'attività. La maggior parte degli approcci per il riconoscimento dell'attività umana utilizzano principalmente tecniche di apprendimento supervisionato. Il principale svantaggio dei metodi supervisionati è la necessità di una quantità significativa di dati etichettati per addestrare un algoritmo di apprendimento automatico. L'etichettatura dei dati per i sistemi di riconoscimento delle attività risulta essere un problema rilevante principalmente per due motivazioni. Innanzitutto, la maggior parte delle tecniche di annotazione richiedono tempo e sono soggette ad errori. In secondo luogo, per ottenere annotazioni affidabili si hanno essenzialmente due possibilità: si può fare affidamento su sensori invasivi come telecamere e microfoni [4] che spesso non sono accettabili per motivi di privacy, oppure, si utilizzano tecniche di annotazione come il metodo di campionamento dell'esperienza [5] (noto anche come un "diario giornaliero" o "valutazione momentanea ecologica", è una metodologia di ricerca longitudinale

intensiva che prevede di chiedere ai partecipanti di riferire quali siano i propri pensieri, sentimenti, comportamenti e/o ambiente, in più occasioni nel tempo), ma risulta essere fastidioso per gli utenti in particolare per l'annotazione di attività a breve termine. Siccome molti problemi pratici, riguardanti la creazione di un modello di attività che possa essere generalizzato per un vasto numero di utenti, risultano essere incentrati sull'elevato costo dell'etichettatura dei dati ed una conseguente facilità di raccolta di un elevato numero di dati non etichettati, è stata proposta come alternativa nella ricerca sull'apprendimento automatico l'apprendimento semi-supervisionato. L'obiettivo finale dell'apprendimento semi-supervisionato è propagare ai dati non etichettati le etichette appartenenti ad altri. In questo modo è possibile sfruttare i dati non etichettati, disponibili in gran quantità, e una piccola porzione di dati etichettati. Siccome molte attività umane di interesse per il riconoscimento vengono eseguite su base giornaliera, è relativamente facile produrre grandi quantità di dati di attività non etichettati. Pertanto, l'apprendimento semi-supervisionato si presta naturalmente al riconoscimento dell'attività.

Una tecnica specifica appartenente al settore dell'apprendimento semi-supervisionato è nota come *active learning*. Tale tecnica ha l'obiettivo esplicito di ridurre la richiesta di dati di addestramento etichettati e pertanto concentrare lo sforzo sull'etichettatura dei dati per rendere il modello più accurato. I dati più utili per migliorare il classificatore di fatto sono quelli più informativi, ovvero sono i dati per cui il classificatore ha rilevato una maggiore incertezza. Dunque, ottenendo un feedback relativo a tali dati, è possibile mitigare l'ambiguità rilevata durante la classificazione. L'utilizzo di tecniche di active learning consiste nel chiedere all'utente un feedback relativo all'attività in svolgimento (o precedentemente eseguita) quando il modello risulta incerto durante la classificazione. In questo modo è possibile chiedere all'utente di etichettare piccole parti di campioni [3] al fine di aggiornare e migliorare il modello di riconoscimento. Inoltre il classificatore è addestrato con pochi dati e l'incertezza viene valutata a partire dallo stream di dati non etichettati, ricevuti come input del sistema, e su cui il classificatore cerca di riconoscere le attività svolte dall'utente. Pertanto l'active learning risulta essere una soluzione ideale per costruire un modello di classificazione efficiente. Tuttavia, spesso, le notifiche ricevute sui dispositivi personali possono risultare molto fastidiose per gli utenti. Questo accade perché gli studi svolti tramite l'utilizzo di tecniche di active learning sono basati su dataset che non considerano il contesto dell'utente per capire quando e se interrompere lo svolgimento di un'attività a causa della ricezione di una notifica, ma semplicemente assumono che l'utente possa rispondere in qualsiasi momento. Questo approccio risulta essere controproducente in sistemi reali sia per il sistema sia per l'utente. Il primo perché potrebbe non ricevere dei feedback oppure riceverli errati, il secondo perché consuma l'attenzione in modo casuale e inefficiente provocando un aumento degli errori e dello stress.

Tale disturbo, generato da un'interruzione provocata da agenti esterni, è possibile riscontrarlo anche all'interno della vita quotidiana. Generalmente, infatti, ogni essere umano nella quotidianità deve svolgere delle piccole analisi di valutazione relativamente allo stato di interrompibilità di altri individui per poter iniziare una conversazione (per interrompibilità si indica quanto un individuo sia disposto ad interrompere l'attività in svolgimento per interagire o dedicarsi ad un'altra attività). Ad esempio, normalmente, i soggetti adulti effettuano questa tipologia di analisi molto rapidamente e con il minimo sforzo poiché si basano sulla propria esperienza pregressa. Questa valutazione consente di bilanciare i vantaggi di un'interruzione con il suo costo e il mantenimento di tale equilibrio solitamente si traduce in un comportamento riconosciuto come socialmente appropriato (o in via informale: civile o educato). Sebbene esistano sistemi automatici per rilevare ed esaminare il contesto, non sono stati mai fatti degli studi per capire quali contesti sono più adatti e quali meno nello specifico ambito dell'active learning. Esistono diversi metodi per eseguire tali rilevazioni ma non ancora incentrati su active learning, o meglio non sono mai state proposte tecniche context-aware per l'active learning principalmente perché gli esperimenti sono fatti su dataset e non all'interno di scenari realistici. Per tale motivo, i sistemi di comunicazione non possono agire in modo simile al comportamento umano ma piuttosto agiscono analogamente in tutte le situazioni, in altre parole non agiscono in modo appropriato secondo la varietà di contesto in cui operano.

Recenti studi confermano che le interruzioni in momenti non opportuni portano ad una disaffezione verso il sistema quando il numero di notifiche risulta essere troppo elevato, ad una riduzione del rendimento dei lavoratori, ad un aumento degli errori e dello stress [18]. Tuttavia, mentre in passato gli individui riorganizzavano i propri orari di lavoro o cercavano luoghi isolati per diminuire il disturbo arrecato da agenti esterni, le tecnologie pervasive oggigiorno rendono difficile sfuggire alle interruzioni indesiderate. Ad incrementare questo stato di pervasività vi è anche la continua nascita di una vasta gamma di applicazioni per smartphone ed altri dispositivi mobili che tende ad aumentare sempre più il contatto con l'utente.

In passato infatti l'importanza del momento in cui avviene un'interruzione è stata riconosciuta in psicologia: analizzando i processi di commutazione dei compiti, i ricercatori hanno dimostrato che le interruzioni provocate dalla ricezione di notifiche sul dispositivo utente arrecano diversi livelli di disturbo in relazione a quale fase di esecuzione dell'attività è stata raggiunta [19]. Per questo motivo è sempre più necessario costruire dei sistemi che interagiscano in modo adeguato con l'utente. Pertanto, l'utilizzo di tecniche di active learning nell'ambito del riconoscimento di attività è molto importante per ottenere informazioni preziose per migliorare il classificatore ma è necessario valutare il *trade-off* tra l'interruzione provocata e il potenziale valore

dell'informazione ottenuta. Per identificare i momenti opportuni in cui il sistema possa inoltrare le notifiche al dispositivo utente è necessario definire cos'è un'interruzione di successo e formulare delle ipotesi verificabili relativamente a ciò che costituisce la relazione stimolo-reazione. Se si considera un sistema tale per cui avviene la ricezione di uno stimolo, sotto forma di notifica, sul dispositivo mobile di un utente, avviene una reazione quando un utente risponde al contenuto fornito attraverso la medesima notifica. Questo definisce un'interruzione di successo poiché la ricezione della notifica, oltre ad aver interrotto l'attività dell'utente, ha provocato la reazione dell'utente, ovvero l'utente fornisce un feedback. Anche se ampia, questa definizione non abbraccia tutti i casi di interruzione mobile, tuttavia assomiglia molto alle interruzioni di e-mail, SMS e social network online.

Questo studio si è incentrato sulla progettazione di una metodologia sperimentale e sullo sviluppo di un prototipo al fine di valutare lo stato di accettabilità dell'utente dell'interruzione provocata dalla ricezione di notifiche di active learning. Per eseguire tale valutazione, si è scelto di utilizzare l'approccio di valutazione HCI “*Human Computer Interaction*” (o in italiano “Interazione Uomo Macchina”, ovvero lo studio dell'interazione tra gli utenti e i computer per la progettazione e lo sviluppo di sistemi interattivi che siano usabili, affidabili e che supportino e facilitino le attività umane) noto come *Wizard of Oz*. Il metodo Wizard of Oz permette di simulare il funzionamento di un sistema di activity recognition basato su active learning senza che l'utente sia a conoscenza che il sistema con cui interagisce non è automatizzato ma interamente pilotato. Tale approccio è di fondamentale importanza all'interno di questa sperimentazione perché, siccome il sistema di activity recognition è soggetto ad errori e il momento in cui si verificano delle query risulta imprevedibile, si vuole essere in grado di controllare interamente la sperimentazione mandando le notifiche di active learning in momenti precisi per valutarne l'accettabilità. Pertanto permette di condurre una valutazione più accurata senza dover fondare la sperimentazione sulla casualità dell'effettivo errore di valutazione di un sistema funzionante. Per eseguire tale valutazione viene analizzato il contesto in cui si trova l'utente sia al momento dell'arrivo della notifica sia al momento dell'invio del feedback da parte dell'utente. Il contesto è descritto in parte dalle proprietà fisiche dell'ambiente, come la posizione, l'attività svolta, il momento della giornata, il tempo meteorologico e la posizione semantica, ma per la valutazione dello stato di accettabilità arrecato dalla ricezione di notifiche di active learning il contesto include anche lo stato interno del soggetto, come ad esempio il suo coinvolgimento in una specifica attività, l'ambiente sociale in cui si trova e le emozioni indotte da tale interruzione. Una volta delineate le metriche necessarie da raccogliere durante la sperimentazione per condurre la valutazione finale, lo studio si è incentrato sullo sviluppo di un prototipo. Per la realizzazione del prototipo si è scelto di analizzare uno specifico caso d'uso ovvero lo svolgimento di

attività in un contesto indoor (più specificatamente è stato considerato come luogo di sperimentazione una smart-home lab all’interno del laboratorio EveryWare Lab). Per condurre la sperimentazione sono state sviluppate due applicazioni, *applicazione wizard* ed *applicazione tablet*, in grado di comunicare tra di loro nel seguente modo: l’applicazione wizard è una WebApp che permette di formulare ed inviare il dubbio durante il riconoscimento di un’attività e permette al wizard di inserire le informazioni di contesto durante le fasi di sperimentazione; l’applicazione tablet è un’applicazione Android (sviluppata all’interno di un precedente lavoro di tesi [7] ed ampliata per questo studio) che permette di ricevere la notifica di active learning e consente all’utente di fornire una risposta che verrà inoltrata all’applicazione wizard. Inoltre tutti i dati raccolti durante la sperimentazione vengono salvati in modo persistente all’interno di un database in modo da permettere l’analisi finale.

I contributi dati nel seguente lavoro di tesi sono i seguenti:

- Progettazione della metodologia sperimentale volta alla valutazione dello stato di accettabilità dell’utente durante l’utilizzo di sistemi di activity recognition basati su active learning mediante l’utilizzo del metodo Wizard of Oz.
- Analisi di un caso d’uso dell’applicazione, in particolare in un contesto indoor all’interno di una simulazione di casa domotica.
- Implementazione del prototipo composto da due applicazioni (*applicazione wizard* e *applicazione utente*) in grado di comunicare tra loro e simulare l’interazione di un sistema di activity recognition basato su active learning.

I capitoli di questo lavoro di tesi sono strutturati nel modo citato di seguito.

Il capitolo 1 presenta lo stato dell’arte relativo all’ambito del riconoscimento di attività descrivendo quali tipologie di attività vengono riconosciute e con quali tecniche si può raggiungere tale obiettivo. In particolare questo studio si è incentrato sull’utilizzo di tecniche di active learning e dei problemi che queste causano. Inoltre viene esposto il metodo di valutazione HCI “*Human Computer Interaction*”, utilizzato all’interno di questo studio, noto come *Wizard of Oz*, spiegando le sue caratteristiche generali. Il capitolo 2 descrive la progettazione dell’esperimento, ovvero la struttura dell’esperimento e le metriche utilizzate per raccogliere informazioni utili al fine di effettuare la valutazione finale sullo stato di accettabilità dell’utente durante la ricezione di notifiche di active learning. Il capitolo 3 spiega la pianificazione dell’esperimento relativa ad un caso d’uso specifico, ovvero lo svolgimento delle attività in un contesto indoor, descrivendo i requisiti che tale sistema deve avere per condurre la sperimentazione. Inoltre viene definito il setup sperimentale utilizzato per questo studio, le fasi di esecuzione dell’esperimento e gli scenari considerati per l’esperimento. Il capitolo 4 descrive nel dettaglio il sistema implementato, spiegando il funzionamento di ogni

componente e il relativo flusso di dati. Infine, nel capitolo 5 si traggono conclusioni sui risultati ottenuti dal lavoro svolto e si analizzano i problemi individuati e i possibili sviluppi per perfezionare il lavoro proposto.

Capitolo 1

Stato dell'arte

Nel seguente capitolo viene data una breve panoramica riguardo il settore denominato *activity recognition* (ovvero “riconoscimento di attività umane”). Tale settore si sviluppa prettamente all’interno dell’area nota come *pervasive computing* (la quale prevede una forma di computazione altamente dinamica e disaggregata in cui i servizi sono forniti da una serie di componenti distribuiti, generalmente sotto forma di microprocessori, che collaborano tra loro) e in particolar modo nel ramo del *context-awareness* (ovvero la capacità di un sistema, o di un componente del sistema, di adattare i propri comportamenti a seguito di una raccolta di informazioni relative all’ambiente in cui opera in un dato momento). Viene spiegato inoltre da quali macro-fasi è costituito il riconoscimento di attività umane e viene data una breve panoramica sulle principali categorie di cui è composto: *vision-based activity recognition*, *sensor-based activity recognition*, *data-driven activity recognition* e *knowledge-driven activity recognition*.

Successivamente vengono esposte quali sono le tipologie di attività che il sistema considera durante la classificazione, distinguendo le attività di alto livello dalle attività di basso livello, e considerando anche l’ambientazione in cui vengono svolte, ovvero ambientazione indoor oppure outdoor. Solitamente i dispositivi usati per interagire con il sistema cambiano in relazione allo scenario considerato (ad esempio un dispositivo tablet verrà più facilmente utilizzato all’interno di un’ambientazione indoor, mentre le cuffie probabilmente vengono più facilmente utilizzate in un’ambientazione outdoor).

Dopodiché viene descritta una tecnica utilizzata all’interno di molti studi per il riconoscimento delle attività nota come *active learning*. Essa prevede che il sistema, durante la classificazione dell’attività in corso d’opera, inoltri una richiesta di feedback all’utente nel caso in cui si verifichi un’ambiguità nella valutazione. Questa tecnica apporta notevoli vantaggi al fine di migliorare l’accuratezza di classificazione nel tempo, considerando lo scenario in cui la quantità di dati etichettati è ridotta,

ma generalmente le tecniche esistenti basate su active learning per il riconoscimento di attività umane non considerano il contesto in cui vengono inoltrate le notifiche al dispositivo dell’utente, o meglio, non sono in grado di valutare quanto la ricezione di una notifica in uno specifico contesto possa infastidire l’utente. Per tale motivo questo studio è incentrato sull’utilizzo di tecniche basate su active learning al fine di progettare un sistema per la valutazione dell’interrompibilità dell’utente.

Nell’ultima sezione viene descritto l’approccio di valutazione HCI “*Human Computer Interaction*” (o in italiano “Interazione Uomo Macchina”), ovvero lo studio dell’interazione tra gli utenti e i computer per la progettazione e lo sviluppo di sistemi interattivi che siano usabili, affidabili e che supportino e facilitino le attività umane, noto come *Wizard of Oz* ed utilizzato durante la sperimentazione. È stato scelto tale metodo poiché permette di pilotare la sperimentazione al fine di determinare quanto l’utente reputi accettabile l’interruzione provocata dall’arrivo di una notifica inoltrata dal sistema.

1.1 Riconoscimento di attività umane

Recentemente, l’ambito di ricerca relativo al riconoscimento di attività umane ha ricevuto una crescente attenzione causata dalle spinte intense degli ultimi sviluppi tecnologici e delle richieste applicative. Infatti, come si evince dall’articolo [6], nell’ultimo decennio le tecnologie dei sensori, in particolare i sensori a bassa potenza, a basso costo, ad alta capacità e miniaturizzati, le reti di comunicazione cablate e wireless ma anche le tecniche di elaborazione dei dati, hanno compiuto progressi sostanziali. I progressi e la maturità di queste tecnologie di supporto hanno spinto i focus di ricerca delle suddette aree a spostarsi dalla raccolta e trasmissione di dati di basso livello verso l’integrazione delle informazioni di alto livello, l’elaborazione del contesto e il riconoscimento delle attività. Allo stesso tempo, le soluzioni per una varietà di problemi del mondo reale sono diventate sempre più dipendenti dal riconoscimento delle attività. È il caso, ad esempio, di smart cities e smart buildings.

Il riconoscimento dell’attività umana (per “attività umane” si intende una serie più o meno articolata di movimenti del corpo che identificano delle azioni precise) risulta essere uno tra i temi di ricerca più promettenti all’interno dell’area denominata *pervasive computing*, ovvero un modello di interazione uomo-macchina (o IUM) in cui l’elaborazione delle informazioni è stata interamente integrata all’interno di oggetti o durante lo svolgimento di attività tipiche della vita quotidiana. Generalmente i dispositivi utilizzati per tale scopo sono connessi in rete e costantemente disponibili e, a differenza del *desktop computing* (il cui paradigma prevede che un utente individuale azioni consciamente una singola apparecchiatura per uno specifico scopo), il *pervasive computing* può avvenire con qualsiasi dispositivo, in qualsiasi momento, in

qualsiasi luogo e in qualsiasi formato di dati su qualsiasi rete e può trasferire compiti da un dispositivo ad un altro come, ad esempio, un utente si sposta dalla sua auto al suo ufficio. In tale ambito possiamo distinguere alcune sotto aree tra cui:

- *mobile computing*;
- sicurezza basata sulla sorveglianza;
- *context-aware computing*, ovvero la consapevolezza del contesto, all'interno delle tecnologie dell'informazione e della comunicazione si riferisce alla capacità di prendere in considerazione il contesto di riferimento di entità, come utenti o dispositivi, non limitandosi alle entità stesse ma, ad esempio, considerando anche la posizione in cui si trovano;
- AAL, ovvero *Ambient Assisted Living*, volto alla creazione di ambienti intelligenti che si adattano in modo indipendente, proattivo e specifico alle esigenze e agli obiettivi dell'utente al fine di consentire alle persone anziane, disabili e/o non autosufficienti di vivere autonomamente in un ambiente privato.

Un esempio di applicazione potrebbe essere l'ambito della sorveglianza e della sicurezza, esse cercano di utilizzare le tecnologie di riconoscimento delle attività al fine di affrontare le minacce dei terroristi, oppure il settore che si occupa di *Ambient Assisted Living* (AAL) mira a sfruttare il monitoraggio, il riconoscimento e l'assistenza delle attività per sostenere la vita indipendente e l'invecchiamento in atto, ma anche altre applicazioni emergenti come la creazione di sale riunioni intelligenti ed anche ospedali intelligenti che dipendono dal riconoscimento dell'attività per fornire interazioni multimediali, servizi proattivi e assistenza personalizzata per le attività considerando il contesto di riferimento. Pertanto il riconoscimento delle attività è un processo complesso che può essere approssimativamente caratterizzato da quattro compiti di base che si possono riassumere nel seguente modo:

1. scegliere e distribuire sensori appropriati per oggetti e ambienti al fine di monitorare e acquisire il comportamento di un utente insieme al cambiamento di stato dell'ambiente;
2. raccogliere ed immagazzinare le informazioni percepite attraverso tecniche di analisi dei dati e/o formalismi di rappresentazione della conoscenza ad appropriati livelli di astrazione;
3. creare modelli di riconoscimento delle attività al fine di consentire ai sistemi/agenti software di condurre classificazioni e manipolazioni dei dati;
4. installazione di tali modelli affinché si possa effettuare il riconoscimento di attività umane in real-time (fase di *deployment*).

In quanto tale, il settore di riconoscimento di attività umane è stato classificato nei seguenti modi:

1. *Vision-based* versus *sensor-based* activity recognition.

In termini di tipologia di sensore utilizzato per il monitoraggio dell'attività, il riconoscimento di attività umane può essere classificato in due categorie.

Il primo è indicato come riconoscimento di attività basato sulla visione. Essa si fonda sull'uso di strutture di rilevamento visivo, come videocamere, per monitorare il comportamento di un soggetto e i cambiamenti ambientali. I dati generati sono quindi sequenze video o dati visivi digitalizzati. Gli approcci in questa categoria sfruttano le tecniche di visione artificiale, tra cui l'estrazione delle caratteristiche, la modellazione strutturale, la segmentazione del movimento, l'estrazione dell'azione e il tracciamento del movimento per analizzare le osservazioni visive per il riconoscimento dei modelli riscontrando però problemi di intrusività e privacy, specialmente in ambienti smart-home.

La seconda categoria è denominata riconoscimento di attività basato su sensori. Essa si fonda sull'uso di tecnologie emergenti di rete di sensori per il monitoraggio dell'attività. In questo caso i dati generati comprendono principalmente serie temporali e/o diversi valori di parametri che vengono solitamente elaborati attraverso la fusione dei dati, metodi di analisi probabilistica o statistica e tecnologie di conoscenza formale per il riconoscimento dell'attività. Tali sensori possono essere collegati a un soggetto sotto osservazione, vale a dire sensori indossabili, come gli smartphone, oppure sensori ambientali. Esempi di sensori indossabili sono unità di misurazione inerziale e tag di identificazione a radiofrequenza (RFID) per raccogliere le informazioni comportamentali di un attore. Oltre i sensori indossabili bisogna considerare anche i sensori ambientali utilizzati principalmente all'interno di case domotiche (*smart-home*). Un esempio di studio per il riconoscimento di attività tramite sensori sono “*Studio di un'interfaccia conversazionale per ambienti smart-home multiutente*” [7] nel caso di un ambiente domestico domotizzato.

Entrambe le categorie sono efficaci per riconoscere i movimenti fisici come ad esempio esercizi fisici, in particolar modo però il riconoscimento di attività umane basato sulla visione è stato al centro della ricerca per un lungo periodo di tempo a causa del suo ruolo importante in aree come la sorveglianza, l'apprendimento dei robot e la sicurezza antiterrorismo.

2. *Data-driven* versus *knowledge-driven* activity recognition.

Sebbene i dati relativi all'attività possano essere ottenuti attraverso il monitoraggio dell'attività stessa (tramite visione o sensori), i modelli di classificazione delle attività sono fondamentali per interpretare tali dati e dedurne una classificazione corretta. Tra i metodi utilizzati dai modelli di classificazione di attività

umane ne esistono due particolarmente noti.

Il primo consiste nell'apprendere modelli di attività da set di dati preesistenti su larga scala dei comportamenti degli utenti utilizzando tecniche di *data mining* e *machine learning*. Questo metodo prevede la creazione di modelli di attività probabilistici o statistici, seguiti da processi di formazione e apprendimento. Poiché questo metodo è guidato dai dati e l'inferenza dell'attività che ne consegue si basa su una classificazione probabilistica o statistica, viene spesso indicato come approccio basato sui dati o *bottom-up*. I vantaggi degli approcci basati sui dati sono le capacità di gestire l'incertezza e le informazioni temporali. Tuttavia, questo metodo richiede grandi set di dati per la formazione e l'apprendimento e soffre della scarsità di dati. È anche difficile applicare modelli di attività appresi da un soggetto all'altro. In quanto tale, questo metodo soffre di problemi di scalabilità e riutilizzabilità.

Il secondo metodo per costruire modelli di attività consiste nello sfruttare una ricca conoscenza pregressa nel dominio di interesse per costruire modelli di attività direttamente utilizzando l'ingegneria della conoscenza e le tecnologie di gestione. Questo di solito comporta l'acquisizione della conoscenza, la modellazione formale e la rappresentazione. I modelli di attività generati con questo metodo vengono normalmente utilizzati per il riconoscimento o la previsione dell'attività attraverso il ragionamento logico formale, ad esempio deduzione, induzione o abduzione. In quanto tale, questo metodo è indicato come approccio basato sulla conoscenza o *top-down*. Gli approcci basati sulla conoscenza hanno il vantaggio di essere semanticamente chiari, logicamente eleganti e facili da avviare. Tuttavia, sono deboli nel gestire l'incertezza e le informazioni temporali e i modelli potrebbero essere visti come statici e incompleti. Un esempio di sistema ibrido semi-supervisionato e basato sulla conoscenza per il riconoscimento delle attività in tempo reale è *CAVIAR* presente nell'articolo [8].

1.2 Caratterizzazione delle attività

Per quanto concerne l'analisi delle attività è necessario distinguere le attività di basso livello, ovvero attività caratterizzate solo da un movimento fisico e quindi più facilmente riconoscibili dal sistema (ad esempio "correre"), dalle attività di alto livello, ovvero attività caratterizzate da una concomitanza di movimenti fisici e quindi più difficilmente riconoscibili perché richiede un sistema più sofisticato (ad esempio "cucinare"). Entrambe le tipologie comprendono delle attività che possono essere svolte sia in contesti indoor che outdoor, ad esempio è possibile svolgere l'azione di "camminare" oppure di "mangiare" sia in luoghi esterni che in strutture al chiuso. Solitamente in letteratura, in base alla tipologia di sensori disponibili, è più comune monitorare le attività di basso livello in ambientazioni outdoor (anche se vengono eseguite

molto in ambientazioni indoor) poiché difficili da riconoscere con sensori indossabili che monitorano movimenti fisici, mentre nelle ambientazioni indoor, grazie all'utilizzo di sensori ambientali, è più facile monitorare le attività di alto livello. È necessario specificare che questo studio si concentrerà prevalentemente su *sensor-based activity recognition* basato su *machine learning* e pertanto verranno utilizzati, coerentemente con lo scenario, sensori indossabili e sensori ambientali.

L'ambientazione indoor comprende l'utilizzo di dispositivi installati e collegati tra loro all'interno di un ambiente domestico, formando quella che viene definita una *smart-home*. Non esiste ancora una definizione ufficiale di *smart-home*, tuttavia è possibile definirla come una struttura dotata di tecnologie domotiche in grado di consentire agli abitanti di programmare, automatizzare e gestire in remoto i dispositivi presenti e fare in modo che questi ultimi monitorino, supportino e migliorino la routine quotidiana di una o più persone. Generalmente gli obiettivi più importanti che vengono posti sono: un miglioramento del comfort, assistenza durante periodi di riabilitazione, monitoraggio di parametri fisici, eccetera. Negli ultimi tempi si è aggiunto anche il task del riconoscimento di attività (ADL - *Activities Of Daily Living*), comprendenti soprattutto le attività di alto livello specificate precedentemente, e il posizionamento degli utenti all'interno della casa, utile per monitorare lo stato di soggetti sensibili quali persone anziane o soggetti con disturbi cognitivi di lieve entità. La maggior parte dei sistemi per il riconoscimento di attività umane all'interno di *smart-home* avvengono tramite sensori integrati in dispositivi mobili. Inoltre tramite i sensori ambientali è possibile estendere il controllo e la gestione dell'ambiente circostante ad esempio monitorando la temperatura, l'umidità e permettendo di pilotare dispositivi quali ventilatori o serrature. Tipicamente quindi le interfacce utilizzate in questo ambiente sono l'interfaccia vocale, per permettere di rispondere agevolmente al sistema anche se l'utente è impegnato fisicamente in altre attività (oltre ad essere più facilmente utilizzabile da utenti che hanno una conoscenza tecnologica limitata), l'interfaccia tablet e l'interfaccia smartwatch.

Per quanto riguarda lo scenario outdoor il riconoscimento delle attività avviene tramite sensori integrati in dispositivi mobili (in particolare vengono prettamente utilizzate le interfacce tramite smartwatch e cuffie). I dispositivi mobili raccolgono continuamente dati da diverse fonti mentre i sensori inerziali (ad esempio accelerometro, magnetometro e giroscopio) trasmettono dati relativi ai movimenti fisici dell'utente. Successivamente questi dati vengono raccolti dal sistema e combinati con i dati in riferimento al contesto in cui è stata svolta la seguente attività attraverso servizi web disponibili pubblicamente (ad esempio la posizione geografica oppure le condizioni metereologiche). Pertanto per "dati di contesto" si indicano principalmente le informazioni sull'ambiente che circonda l'utente durante lo svolgimento di un'attività.

1.3 Tecniche di active learning e il loro utilizzo

Come anticipato nella sezione precedente, il riconoscimento di attività umane è un settore del *context-awareness* (la quale è una branca del *pervasive computing*) che tramite l’acquisizione di informazioni, da sorgenti di diverso tipo, mira a studiare tecniche le quali permettono ad una macchina di apprendere in modo autonomo un metodo di classificazione delle attività umane in base al grado di complessità.

Esistono diversi algoritmi basati su approcci statistici, tipicamente vengono utilizzate le tecniche di *machine learning* supervisionate, ovvero tecniche che richiedono una grande quantità di dati etichettati con i quali addestrare l’algoritmo al fine di ottenere un classificatore che possa a sua volta etichettare i nuovi dati ricevuti in input. Purtroppo però queste risultano essere svantaggiose per diversi aspetti: richiedono un’importante quantità di dati da diversi soggetti; l’acquisizione di un dataset affidabile, oltre ad essere costoso a livello di tempo, è intrusivo in termini di privacy (siccome è necessario monitorare dei soggetti per lungo tempo); costoso anche in termini finanziari per produrre una grande campagna realistica. Spesso quindi risulta essere impraticabile in scenari realistici in cui i soggetti devono svolgere le attività in modo naturale per un periodo lungo. Per ovviare queste problematiche generalmente vengono utilizzate delle tecniche di apprendimento miste note come semi-supervisionate che permettono di inizializzare il classificatore con un set limitato di dati annotati e dopodiché si cerca di fornire ai dati non annotati delle etichette in modo semi-automatico, tra cui l’*active learning*. L’active learning si occupa di analizzare le attività ricevute in output da un sistema di *machine learning* e permette di interrogare in modo interattivo l’utente per risolvere l’incertezza del sistema nel caso in cui vi fosse la presenza di ambiguità durante la classificazione dell’attività svolta. Queste tecniche di apprendimento semi-supervisionate prevedono quindi l’utilizzo di un training set di dimensioni minori e successivamente, al fine di aggiornare e raffinare le predizioni del classificatore, viene richiesto l’intervento degli utenti i quali possono fornire dei feedback riguardanti le attività in corso attraverso l’ausilio di un’interfaccia. L’intervento dell’utente è richiesto affinché sia possibile aggiornare il modello quando il sistema di *machine learning* effettua predizioni con una confidenza inferiore ad una soglia prestabilita. In questo modo è possibile migliorare il modello di classificazione nel tempo e pertanto rinforzarlo al fine di riconoscere attività sempre in maniera più accurata. Un esempio di recente studio relativo all’uso di tecniche di active learning compatibilmente con lo stato di accettabilità dell’utente è il sistema *CrowdAct* [9]. In tale studio viene proposto un nuova tipologia di active learning ovvero *gamification dell’active learning* e il rilevamento di imprecisioni per l’etichettatura dei dati in crowdsourcing per un sistema di riconoscimento delle attività che utilizza il rilevamento mobile (CrowdAct).

1.4 Problemi riguardo l’usabilità di sistemi basati su active learning

Spesso i lavori realizzati all’interno dell’ambito di riconoscimento di attività umane basati su active learning mostrano dei risultati su dei dataset già raccolti e quindi, siccome non sono mai stati testati in un vero deployment, non considerano il reale utilizzo di queste interfacce da parte dell’utente. O meglio, tali tecniche vengono sfruttate rimanendo quasi completamente ignare del mondo umano in cui operano. Questo accade poichè assumono che gli utenti siano sempre disponibili a fornire un feedback non appena il classificatore è incerto, indipendentemente da cosa stiano facendo in quel momento e da quale sia il contesto in cui si trovano. Ad esempio il sistema potrebbe richiedere un feedback all’utente mentre sta guidando oppure cucinando o mentre sta svolgendo altre attività che richiedono un’attenzione rilevante da parte del soggetto che la sta compiendo. Per questo motivo in genere questi lavori non hanno modo di tenere conto dell’interrompibilità dell’utente. Per interrompibilità si indica la misura che determina quanto un qualsiasi utente sia disposto ad interrompere l’attività in fase di svolgimento per interagire con l’applicazione e quindi fornire un feedback in grado di far apprendere il sistema e risolvere l’incertezza.

I sistemi volti al riconoscimento di attività umane sviluppati tramite algoritmi di active learning agiscono, quindi, essenzialmente in tutte le situazioni in modo equivalente, non adeguandosi al contesto di riferimento. Se non controllati, i sistemi attuali possono facilmente disturbare o infastidire, consumando la preziosa risorsa dell’attenzione umana in modo casuale e inefficiente, ad esempio portando ad una riduzione del rendimento dei lavoratori oppure ad un aumento degli errori nell’attività che l’utente sta svolgendo e dello stress. Di conseguenza, poichè le tecnologie pervasive richiedono un’interazione sempre maggiore con l’utente ed essendo quindi difficile non causare dalle interruzioni indesiderate, spesso l’utente sceglie di evitare la troppa interazione con il sistema costringendo le interfacce a rimanere silenziose e passive finché non viene richiesto oppure suscitare una possibile disaffezione verso il sistema e quindi proprio l’abbandono dello stesso se venissero fatte troppe query o troppe volte in momenti inopportuni. Se vi fossero degli stimatori relativamente solidi dell’interrompibilità, potremmo migliorare l’esperienza utente in diversi modi rendendo le persone più soddisfatte ed appagate dall’utilizzo di queste interfacce.

In questo progetto si vuole studiare il grado di interrompibilità dell’utente per andare a valutare l’accettabilità di sistemi di riconoscimento di attività umane basati su active learning all’interno di diversi scenari comprendenti sia attività di basso livello che attività di alto livello. In questo modo il feedback fornito dagli utenti permette di migliorare il sistema nel determinare il grado di interrompibilità dell’utente in un dato istante e in un dato contesto.

1.5 Wizard of Oz

Per andare ad analizzare quali sono i momenti e i contesti in cui l’utente ritiene accettabile l’interazione con un sistema di active learning, si è deciso di utilizzare per questo studio il metodo *Wizard Of Oz*. Il metodo *Wizard of Oz* consente di ricavare dallo svolgimento della sperimentazione dei dati maggiormente accurati rispetto ad un normale utilizzo del sistema. Inoltre la caratteristica principale di questo approccio consiste nel far percepire ai soggetti impegnati nella sperimentazione che il sistema sia funzionante quando in realtà l’interazione viene mediata da un operatore umano (chiamato *wizard*) che comanda l’interfaccia a sua insaputa. In questo modo non viene testata l’efficacia del funzionamento dei sensori integrati nei dispositivi ma piuttosto l’efficacia dell’utilizzo di tali dispositivi da parte dell’utente oppure la loro combinazione. Ciò significa che tramite questo metodo è possibile affinare degli aspetti di progettazione in relazione a parti già funzionanti per testare la validità del sistema oppure è possibile analizzare e valutare l’utilizzo di tali dispositivi anche se non ancora creati o se non perfettamente funzionanti al fine di determinare le peculiarità di ciò che dovrà essere affinato o costruito.

L’approccio di prototipazione del metodo *Wizard of Oz* (WOz), ampiamente utilizzato nella ricerca sulla *Human Computer Interaction*, è particolarmente utile nell’esplorazione delle interfacce utente per sistemi pervasivi, ubiquitous (“*ubiquitous computing*”, in italiano “computazione ubiqua”), è un modello post-desktop di interazione uomo-macchina o HCI, in cui l’elaborazione delle informazioni è stata integralmente integrata all’interno di oggetti ed attività di tutti i giorni) o di realtà mista che combinano il rilevamento complesso e la logica di controllo intelligente. Il vasto spazio di progettazione per tali interfacce non tradizionali offre molte possibilità per l’interazione dell’utente attraverso una o più modalità e spesso richiede complesse implementazioni hardware e software. Il metodo WOz aiuta i progettisti a evitare di rimanere bloccati in un particolare progetto o di lavorare sotto una serie errata di presupposti sulle preferenze dell’utente, perché consente loro di esplorare e valutare i progetti prima di investire il considerevole tempo di sviluppo necessario per costruire un prototipo completo. Come con altri strumenti utilizzati solo durante la fase di progettazione (detti *throwaway tools* ovvero “strumenti usa e getta”), la maggior parte delle interfacce create tramite il metodo WOz non sono concepite per evolversi con il sistema. Quindi i progettisti tendono ad utilizzare questo approccio una (o probabilmente due) volte durante l’evoluzione di un sistema, in netto contrasto con altri metodi di valutazione che potrebbero essere utilizzati ripetutamente durante l’evoluzione del sistema. Nonostante la poca elasticità di questo approccio, gli studi effettuati con il metodo WOz possono aiutare i progettisti a valutare sistemi parzialmente completi ma anche a far progredire il design della tecnologia sottostante.

Uno dei motivi per l'uso limitato del metodo WOz è lo sforzo richiesto per progettare un'interfaccia di successo attraverso questo approccio ed integrarla con un sistema incompleto.

La progettazione di sistemi interattivi in genere implica un processo iterativo di brainstorming, prototipazione, sviluppo, test utente e valutazione. Questo non è un processo chiaro, spesso viene iterato attraverso molti cicli prima di raggiungere un sistema finale. La **Figura 1** illustra come si può utilizzare WOz durante il processo di progettazione, man mano che un sistema si evolve. Di fatto il ciclo di sviluppo è molto più complicato e comprende anche la possibilità che la visione del sistema cambi in vari punti, alterando la curva di sviluppo tecnologico. I punti blu e rossi nella **Figura 1** rappresentano esempi di lavori precedenti con il metodo WOz.

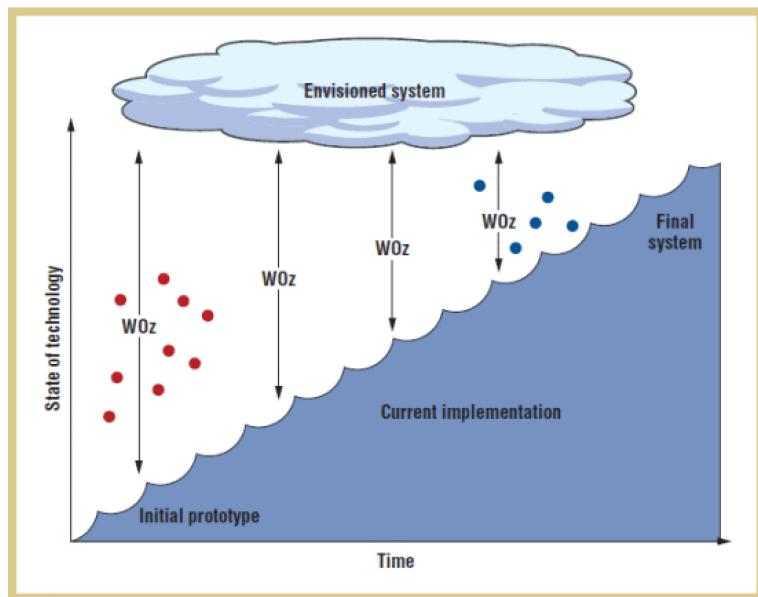


Figura 1: Una visione generale dell’evoluzione del design e del ruolo che il metodo WOz può svolgere nel facilitare la valutazione di un’esperienza immaginata [17]. Il metodo WOz è stato tradizionalmente utilizzato all’inizio per simulare la tecnologia non sviluppata (esempi indicati come punti rossi) o verso la fine della progettazione, quando la procedura guidata può supervisionare un’implementazione completa o simulare una piccola parte della tecnologia (punti blu). È possibile anche usare il metodo WOz nella parte centrale dello sviluppo come mezzo per passare a un progetto finale.

Negli studi WOz, un *wizard operator* svolge generalmente un ruolo in un sistema informatico work-in-progress: potrebbe simulare i dati dei sensori, le informazioni contestuali o l'intelligenza del sistema. Si può invece usare la parola complementare *puppet* per fare riferimento alla simulazione dell'esperienza utente pilotata dal wizard. Durante una valutazione basata su WOz, il wizard diventa essenzialmente un sostegno per simulare l'interfaccia e le interazioni previste prima che il sistema funzioni completamente. Con il progredire dello sviluppo della tecnologia, il divario colmato dal wizard si riduce.

Tradicionalmente, un *wizard operator* svolge uno tra i seguenti due ruoli:

- un *controller* che si occupa di simulare completamente un componente di sistema non costruito (quindi si occupa di simulare un sensore o l'intelligenza del sistema),
- oppure un *supervisor* che supervisiona un sistema funzionante e, se necessario, annulla le decisioni prese dal sistema o dall'utente.

Un ruolo meno comune invece è quello di *moderator* che si colloca tra i ruoli di controllore e supervisore. Quando una tecnologia o un componente di sistema funziona ma non è ancora attendibile invece di collegarlo al sistema si può assumere il ruolo di moderatore per osservare l'output di questo componente e quindi inviare tale output come input al resto del sistema. Tuttavia, poiché il moderatore può sovrascrivere e quindi modificare l'output del componente o del sensore prima che raggiunga il resto del sistema, una “valutazione moderata” può comunque fornire l'esperienza utente immaginata completa. Ciò consente di ottenere utili feedback di valutazione mentre si raccoglie un ricco set di metadati per valutare lo sviluppo del componente di sistema o del sensore in questione. Utilizzando il ruolo di moderatore è possibile eseguire una valutazione molto più realistica nonostante il sistema sottostante si comporti in modo inaspettato (ad esempio comportandosi in modo irregolare o fallendo completamente). Analizzando però ciò che stava accadendo quando la procedura guidata si è discostata dalle decisioni del sistema, la valutazione potrebbe anche scoprire false ipotesi di progettazione. Per questo condurre degli studi WOz nelle fasi intermedie dello sviluppo del sistema aiuta a perfezionare l'interazione dell'utente consentendo allo stesso tempo lo sviluppo della tecnologia. In particolar modo se si raccolgono dati sull'interazione dell'utente e sull'operatore della procedura guidata durante l'esperienza moderata, è possibile eseguire una valutazione molto più realistica nonostante il sistema sottostante si comporti in modo imprevisto (probabilmente perché si comporta in modo irregolare o fallisce completamente).

Le tecniche WOz sono state utilizzate in vari modi, a seconda delle tecnologie disponibili e degli obiettivi specifici del progetto. Il suo ruolo più comune nello sviluppo

della *Human Computer Interaction* attribuisce alla procedura guidata la responsabilità di controllare completamente un pezzo mancante dell’infrastruttura tecnologica (dalla simulazione dell’intero sistema alla simulazione di un solo sensore). Utilizzato all’inizio della progettazione come metodo di prototipazione flessibile, il metodo WOz presenta agli utenti delle “bozze” di interfaccia anche quando non è chiaro quale dovrebbe essere la tecnologia sottostante. Nello studio di Scott Hudson e i suoi collaboratori [10] l’intento era quello di capire quale fosse la disponibilità degli utenti ad essere interrotti in un ambiente d’ufficio. Il loro studio WOz li ha aiutati a identificare i sensori appropriati per lo spazio. Stephen Voida e i suoi colleghi [11] invece hanno utilizzato WOz per studiare le tecniche di interazione di base per un ambiente di realtà aumentata basato su proiettori o telecamere. Il loro intento era quello di comprendere il comportamento dell’utente e le modalità di interazione preferite non vincolate da limitazioni imposte dalla tecnologia, come guanti speciali o movimenti altamente vincolati per aiutare un sottosistema di visione artificiale.

Più avanti nel processo di progettazione, dopo aver identificato le tecnologie appropriate, queste potrebbero risultare ancora troppo difficili da implementare specialmente per testare problemi speculativi dell’interfaccia utente. Nello strumento Topiaria [12], il metodo WOz svolge il ruolo di sensore di posizione (ad esempio, un GPS) durante la progettazione di applicazioni basate sulla posizione. Nella valutazione di Kent Lyons [13] del discorso a duplice scopo, il metodo WOz ha permesso di esplorare una porzione più ampia dello spazio di progettazione analizzando l’input vocale senza restrizioni per i principianti. Quan Tran e i suoi colleghi [14] hanno utilizzato un wizard per simulare la tecnologia di visione durante lo sviluppo del Cook’s Collage, un ausilio per la memoria per gli anziani in cucina. Utilizzando WOz, prototipi come questo possono maturare in applicazioni sofisticate che aiutano i ricercatori a testare le teorie sull’interazione senza sprecare tempo nella progettazione di un complesso sistema sottostante che potrebbe non essere necessario nel prodotto finale. Anche nelle applicazioni finite in cui il sistema svolge la maggior parte del lavoro, un approccio come il WOz può svolgere una varietà di ruoli. Ad esempio in Alice’s Adventures in New Media [15], un wizard simula un riconoscimento di gesti come input per un sofisticato motore narrativo basato su agenti come parte di un’esperienza di realtà aumentata.

Tutti questi esempi posizionano il *wizard operator* come *controller* ed utilizzano tale metodo sia all’inizio che alla fine del processo di progettazione. Nella performance in realtà mista (Desert Rain [16]), il wizard interpreta il ruolo di supervisore, aiutando i partecipanti nell’esperienza caso per caso. Il wizard può anche svolgere un ruolo dedicato, aggiungere intelligenza oltre le attuali possibilità di elaborazione o semplicemente monitorare un’esperienza per aiutare quando sorgono problemi.

In questo studio viene utilizzato il metodo WOz all'inizio della sperimentazione: il wizard interpreta il ruolo di *controller* poiché simula il funzionamento di un sistema automatico e dei sensori per il riconoscimento di attività fisiche svolte dall'utente. In questo modo è possibile andare a valutare la reazione dell'utente per stabilire il grado di interrompibilità dell'attività che sta svolgendo. L'interruzione è causata dalla ricezione di una notifica inoltrata dal sistema e programmata dal wizard, poichè stabilirà in quali contesti inviarla al dispositivo utente per effettuare uno studio più approfondito.

Pertanto, all'interno di questo studio, questo approccio permette di simulare l'interazione con un sistema di riconoscimento di attività umane con active learning mentre l'utente sta svolgendo delle attività, o meglio all'utente sembrerà di interagire con un sistema automatizzato ma in realtà il sistema viene gestito da un operatore il quale fa credere al soggetto che il sistema non sia stato in grado di riconoscere correttamente l'attività e di conseguenza necessita di un feedback diretto per risolvere l'ambiguità. In questo modo è possibile raccogliere dati come base per lo studio sull'accettabilità dei sistemi durante lo svolgimento delle attività degli utenti focalizzando l'attenzione prettamente sull'interfaccia utente e senza dover fondare la sperimentazione sulla casualità dell'effettivo errore di valutazione di un sistema funzionante, siccome non è possibile sapere quando un effettivo sistema di activity recognition e inoltre le richieste di feedback, dovute all'uso dell'active learning, sul dispositivo utente e se queste vengano inoltrate realmente.

Capitolo 2

Metodologia sperimentale proposta

In questo capitolo viene descritto il processo progettuale per definire come svolgere l'esperimento per la valutazione dello stato di accettabilità dell'utente relativo all'interruzione causata dall'uso di tecniche di active learning (esse prevedono l'invio sul dispositivo utente di richieste di feedback per risolvere l'ambiguità relativa all'attività, possibilmente in svolgimento). Per condurre la sperimentazione, e pertanto anche la valutazione finale, è necessario come prima fase di tale processo definire quali sono gli obiettivi finali di questo studio, fondamentale per poter strutturare passo passo la sperimentazione. Una volta definiti gli obiettivi che si vogliono raggiungere a fine sperimentazione, viene spiegato come il metodo *Wizard of Oz*, descritto nella sezione 1.5, viene utilizzato all'interno di questo studio. In questa fase vengono esposti quali sono le mansioni del wizard durante la sperimentazione al fine di ricavare le informazioni necessarie per condurre la valutazione finale. È importante evidenziare inoltre che il wizard, durante l'esecuzione dell'esperimento, potrà essere affiancato da collaboratori o volontari. Questo permetterà di avere una maggior accuratezza durante il monitoraggio dell'utente, soprattutto nelle ambientazioni outdoor poiché il wizard deve collocarsi ad una distanza tale da non compromettere il realismo dell'esperimento, ma possono anche fungere da attori all'interno della sperimentazione al fine di simulare degli scenari di tipo sociale (comprendente o la sola presenza di persone oppure un'interazione sociale).

Successivamente vengono descritte quali sono le fasi della sperimentazione in relazione al ciclo di vita di una query. Per *query* si intende il messaggio di richiesta di feedback mandato dal sistema, in cui si propone all'utente di effettuare una scelta tra le attività che il sistema (in questo studio si fa riferimento al wizard) ritiene essere tra le più probabili rispetto all'attività che sta svolgendo in un dato istante e visualizzato sul dispositivo utente. In questo studio però, siccome viene utilizzato il metodo WOz, l'utente crede di utilizzare un sistema che lo monitori durante lo svolgimento di un'attività e che lo interrompa se rileva un'ambiguità durante la classificazione ma

in realtà è il wizard che osserva l'utente, decide quando interromperlo, formula e poi inoltra il messaggio all'interfaccia utente. Pertanto per ogni fase del ciclo di vita di una query il wizard è tenuto a compilare delle informazioni necessarie per la valutazione finale, ad esempio deve selezionare quali attività andranno a formare le opzioni di scelta della query. Nel caso in cui l'utente non fornisca un feedback, il sistema indicherà tali campi come nulli e giudicherà tale scenario altamente interrompibile.

Per valutare il successo dell'interruzione (considerando che si verifica un successo nel momento in cui l'utente fornisce un feedback in un lasso di tempo accettabile) è necessario stabilire delle metriche. Esse comprendono la presenza di risposta, il tempo di latenza dall'arrivo della notifica all'invio del feedback, il sentimento provato dall'utente al momento dell'interruzione ma anche le metriche relative al contesto. Per contesto non si indicano solamente le informazioni caratterizzanti l'ambiente che circonda l'utente durante lo svolgimento di un'attività (ad esempio la posizione in cui si trova) ma si considera un'accezione più ampia comprendente diversi aspetti che influenzano l'interrompibilità dell'utente. Ad esempio si considera se l'utente si trova sui mezzi pubblici (valutazione della posizione semantica) oppure delle condizioni metereologiche in quel dato istante, tali fattori potrebbero infatti influenzare la disponibilità dell'utente nel fornire un feedback al sistema.

Dopodichè la fase successiva della progettazione ha previsto la decisione di un set di attività che l'utente potesse svolgere durante la sperimentazione, distinguendo le attività di basso livello (attività caratterizzate solo da un movimento fisico e quindi più facilmente riconoscibili dal sistema) dalle attività di alto livello (attività caratterizzate da una concomitanza di movimenti fisici, pertanto più difficilmente riconoscibili dal sistema) e le tipologie di informazioni di contesto necessarie considerando l'ambientazione in cui vengono eseguite, ovvero ambientazione indoor e ambientazione outdoor. Ad esempio per l'ambientazione indoor si considera l'informazione di contesto relativa alla posizione semantica con una granularità maggiore rispetto all'ambientazione outdoor (ovvero si analizza, ad esempio, in quale stanza si trova l'utente).

In seguito la progettazione si è focalizzata su come impostare lo svolgimento dell'esperimento ovvero come comunicare all'utente che tipologia di attività svolgere durante la sessione sperimentale. Un altro criterio di giudizio per valutare lo stato di interrompibilità è dato dalla definizione dei momenti in cui è più opportuno inoltrare le notifiche dal dispositivo wizard all'interfaccia utente. Questo significa che per valutare quali sono le condizioni per cui l'utente ritiene accettabile un'interruzione provocata dal sistema, dipende anche da quando il dispositivo utente riceve le notifiche e pertanto quando l'utente viene interrotto rispetto all'esecuzione dell'attività. Il wizard quindi può decidere se inoltrare le notifiche al dispositivo utente mentre il soggetto sta svolgendo l'attività, oppure ad attività conclusa oppure ancora quando l'utente si trova in una situazione di riposo (come ad esempio stare seduto o stare in piedi). Questo aspetto è fondamentale per cercare di rendere la sperimentazione più ampia, ovvero

considerando un'ampia varietà di scenari per cercare di avere una valutazione il più ravvicinata alla realtà, e pertanto studiare qual è la reazione dell'utente al variare del momento in cui si verifica un'interruzione.

Nella fase successiva viene fatta una riflessione riguardo alla tipologia di prompt visualizzato (il wizard dovrà occuparsi di indicare quali sono le attività di opzione tra cui l'utente potrà scegliere ed una volta fornito il feedback l'utente dovrà fornire una valutazione del sentimento attraverso indicando il proprio grado di soddisfazione del servizio), quali interfacce l'utente ha a disposizione per fornire un feedback al sistema (ad esempio l'interfaccia touch e l'interfaccia vocale) e quali possano essere i dispositivi usati per fornire il proprio feedback al sistema in base allo scenario considerato (ad esempio smartphone, tablet, cuffie, smartwatch).

Infine viene definito il target di utenti che necessita tale sperimentazione e l'importanza di condurre nella fase conclusiva una valutazione post-sperimentazione che richieda un parere diretto dell'utente relativamente alla sua esperienza durante l'utilizzo del sistema. In questo modo oltre ai dati ricavati dalla compilazione semi-automatica dei campi durante il ciclo di vita della query (alcuni verranno compilati manualmente dal wizard ed altri automaticamente dal sistema, come ad esempio il timestamp relativo al momento in cui è stata inviata la notifica) l'utente fornisce una valutazione personale e complessiva del sistema permettendo ai progettisti di affinare il sistema.

2.1 Obiettivi dell'esperimento

Al fine di eseguire una valutazione esaustiva giudicando quali siano i momenti più opportuni in cui il sistema possa inoltrare delle notifiche al dispositivo utente, è necessario definire gli obiettivi finali da raggiungere tramite questa sperimentazione e successivamente la tipologia di informazioni che il sistema deve memorizzare affinché si possa eseguire tale analisi. È necessario specificare che la notifica è composta da una query formulata dal wizard durante la sperimentazione (per *query* si intende il messaggio di richiesta di feedback mandato dal wizard, che simula il comportamento di un sistema di activity recognition con active learning, per risolvere l'ambiguità relativa al riconoscimento di un'attività possibilmente in svolgimento). Tale query comprende la visualizzazione di 3 opzioni tra cui l'utente può scegliere per fornire il feedback, ovvero la selezione tra due attività, definite dal wizard al momento dell'invio della query, oppure la scelta dell'opzione alternativa se le prime due non hanno dato la possibilità di fornire l'attività corretta.

Gli obiettivi principali di questo studio comprendono:

- stabilire quali contesti sono classificabili maggiormente interrompibili e se, in tali contesti, è preferibile inoltrare la notifica durante lo svolgimento di un'attività

oppure una volta terminata, ad esempio si può ritenere che l’utente sia più disposto a fornire un feedback al sistema se l’attività considerata è “guardare la tv” piuttosto che durante lo svolgimento dell’attività “andare in bicicletta”;

- stabilire qual è la soglia di latenza accettabile, ovvero se il tempo trascorso tra l’invio della notifica e la ricezione della risposta è superiore ad una soglia prestabilita allora l’interruzione è considerata non appropriata, ad esempio la scorretta definizione della soglia comporta che il soggetto avrà una maggior probabilità di non essere più in grado di ricordarsi l’attività svolta al momento dell’arrivo della query e pertanto, se l’individuo non ha fornito un feedback al momento della ricezione della notifica, molto probabilmente era un momento da considerare non opportuno per l’invio della query;
- stabilire se il cambiamento di contesto tra la ricezione della notifica e l’invio della seguente risposta determina un fattore rilevante per giudicare l’interrompibilità dell’utente, ad esempio se l’attività in corso è “correre” probabilmente l’utente sarà più disposto a fornire un feedback al sistema quando effettua una pausa oppure a conclusione dell’attività (anche se riferita all’attività precedente);
- stabilire se l’utente ritiene accettabile il momento scelto dal sistema (si ricorda che in questo studio è scelto dal wizard) per inoltrare la query al dispositivo utente (tale obiettivo viene raggiunto tramite la richiesta di un feedback esplicito con la visualizzazione dell’interfaccia dedicata alla selezione del sentimento);
- stabilire il numero di notifiche massimo che il sistema può accumulare all’interno di una coda e relative alle attività svolte in un arco temporale predefinito (in questo modo si cerca di evitare che la query faccia riferimento ad un’attività svolta troppo tempo addietro e che il soggetto non riesce a ricordare), ad esempio se l’utente riceve un numero troppo elevato di richieste di feedback potrebbe non fornire una risposta a tutte oppure fornire una o più risposte scorrette al fine di velocizzare la compilazione;
- stabilire, relativamente ad una coda di notifiche, l’intervallo di tempo massimo trascorso dal reale svolgimento della specifica attività considerata all’interno della query e il momento in cui viene effettivamente inoltrata sull’interfaccia, se tale intervallo risulta essere troppo ampio il feedback dell’utente potrebbe essere errato oppure potrebbe non essere fornito a causa della dimenticanza;
- stabilire quanto tempo deve trascorrere tra l’invio di una notifica e la successiva poiché si suppone che l’utente possa essere infastidito dall’arrivo sequenziale di notifiche se all’interno di un arco di tempo piccolo.

Si consideri però che l'ultimo obiettivo, ovvero “stabilire quanto tempo deve trascorrere tra l'invio di una notifica e la successiva per non infastidire l'utente” è un obiettivo molto importante per questo studio ma al contempo difficile da determinare. Tale impedimento è dovuto alla durata dell'esperimento poiché con il metodo WOz non è possibile svolgere esperimenti su un arco di tempo ampio ma solo brevi sessioni in cui l'utente svolge poche attività, pertanto questa specifica modalità sperimentale non permette di acquisire statistiche che possano valere sul lungo periodo e in grado di stabilire più precisamente quale sia la soglia di tollerabilità dell'utente. Ragion per cui all'interno di questo studio non sarà possibile determinare una soglia temporale per l'invio delle notifiche successive ma rimane un aspetto molto importante da evidenziare per stabilire lo stato di accettabilità dell'interruzione da parte dell'utente.

È interessante inoltre osservare come cambia la reazione dell'utente all'interruzione provocata dal sistema (o meglio dal wizard) al variare di questi parametri. Ad esempio se viene utilizzata la coda di notifiche e questa risulta essere eccessivamente lunga, l'utente potrebbe fornire una risposta solo alle prime richieste oppure nel caso di una query riguardante un'attività che non ricorda potrebbe non fornire la risposta. Un altro esempio che si potrebbe prendere in considerazione è se l'utente, invece di non rispondere o di scegliere l'opzione alternativa che comprende la selezione di “nessuna delle due”, fornisca un feedback sbagliato e sarebbe interessante anche analizzare come si comporta l'utente nel caso in cui il sistema fornisca delle alternative che non comprendono l'attività corretta.

Si potrebbe inoltre considerare a seconda del contesto in cui si trova l'utente se l'errore nel fornire l'attività corretta sia dato da un eccessivo stress (ad esempio se l'utente sta correndo, oppure è in metropolitana, etc) o se il ricordo dell'attività non era chiaro o anche se il numero di notifiche era troppo eccessivo e quindi troppo laborioso da poter rispondere singolarmente con accuratezza. Quest'ultimo aspetto riguardante la correttezza del feedback fornito dall'utente è di difficile stima se non all'interno di uno studio basato sul metodo WOz poiché è grazie a questo approccio, ovvero grazie alla stima effettuata in tempo reale dal wizard che supervisiona l'utente durante la sperimentazione, si è in grado di valutare se la risposta fornita dall'utente sia effettivamente l'attività corretta. Bisogna però considerare che i sistemi di active learning cercano di considerare l'ipotesi in cui le labels fornite possano essere “rumorose”, ovvero errate quando l'utente potrebbe fornire un feedback sbagliato.

2.2 Descrizione dell'esperimento basato su WOz

Come anticipato nelle precedenti sezioni, questo studio prevede lo svolgimento di un esperimento basato sul metodo Wizard of Oz. Tale approccio (come spiegato all'interno della sezione 1.5) consente di simulare un sistema di activity recognition basato su active learning al fine di incentrare l'attenzione non sul funzionamento del sistema bensì sulle informazioni ricavate da questo per effettuare una valutazione finale più accurata. Per questo motivo, al fine di rendere la sperimentazione più realistica, l'utente interagirà con il sistema credendo che le notifiche per risolvere l'incertezza del sistema sull'identificazione dell'attività siano generate automaticamente dal sistema ma in realtà l'intero sistema è pilotato dal wizard. La sperimentazione consiste nella riproduzione di scenari tipici della vita quotidiana in cui l'utente svolge delle attività in ambientazione casilinga oppure di lavoro o anche di svago (ad esempio al parco). Il wizard durante la sperimentazione osserva l'utente svolgere le attività e decide come impostare e quando inviare la query di active learning sul dispositivo utente. Per far sì che la sperimentazione non venga gestita da un singolo individuo, il wizard verrà affiancato da collaborati o volontari che lo aiuteranno sia per monitorare l'utente durante lo svolgimento di una o più attività ma potrebbero anche essere coinvolti come attori. In questo modo sarà possibile anche considerare e simulare degli scenari di vita quotidiana in cui l'utente si trovi a svolgere un'attività con la presenza di altre persone oppure durante un'interazione sociale.

Tale sperimentazione prevede che il wizard invii un numero cospicuo di notifiche per valutare come l'utente reagisce al variare dello scenario considerato, ma bisogna anche evidenziare che i sistemi basati su active learning (durante un reale utilizzo) hanno la caratteristica di generare una percentuale bassa di query e pertanto difficilmente l'utente, durante l'utilizzo di tale sistema all'interno della vita quotidiana, riceverà un numero consistente di richieste di feedback. Per questo motivo sarà difficile la formazione di una coda di query riferite alla medesima attività soprattutto a seguito di precedenti feedback forniti dall'utente (viceversa potrebbe essere ammissibile se le query all'interno della coda rimandassero ad attività differenti). È necessario quindi spiegare all'utente che in un contesto reale i sistemi di active learning non richiedono un numero elevato di feedback al soggetto (indipendentemente se queste vengano oppure non vengano accumulate) ma all'interno di questa sperimentazione la quantità di query risulterà essere maggiore. Pertanto l'utente deve essere consapevole che il sistema in uso durante la sperimentazione richiederà una quantità consistente di feedback a scopo di studio esplorativo, viceversa, se fosse utilizzato in un contesto reale le notifiche verrebbero inviate in quantità inferiore. In questo modo sarà più semplice avere una maggiore collaborazione da parte dell'utente e reperire dei dati più realistici senza dare eccessivo peso al quantitativo di notifiche inoltrate dal sistema.

Inoltre è possibile considerare due tipologie di target di utenti per condurre la sperimentazione:

- **target generico**, in cui si prendono in considerazione dei soggetti che riescano a coprire il più ampio spettro possibile di età e quindi permettono di effettuare la sperimentazione su una scala di soggetti più estesa ma meno approfondita;
- **target specifico**, in cui si considerano dei soggetti appartenenti ad una fascia di età specifica eliminando l'ampia varietà di casistiche e semplificando l'analisi dell'esperimento.

Per questo studio si preferisce considerare un target di riferimento specifico selezionando dei soggetti di età compresa tra i 18 e i 60 anni. In questo modo si cerca idealmente di non avere dei risultati con tendenze specifiche a riferimento di una specifica fascia di età ma di ottenere dei risultati medi escludendo però le fasce di età estreme. In questo modo è possibile considerare minime le difficoltà causate dalla modalità di interazione con la tecnologia (smartwatch e cuffiette) da parte di bambini o anziani ma anche di evitare dei problemi nella valutazione dell'esperimento poiché risulterebbe di maggior complessità capire se ad esempio il feedback è stato fornito sbagliato per problemi di interrompibilità o per problemi legati al soggetto stesso (ad esempio legati al multitasking, alla tipologia di attività da svolgere, alla memoria).

2.3 Caratterizzazione del contesto

Oltre alle metriche viste precedentemente, per analizzare l'adeguatezza di un'interruzione, è necessario valutare il contesto in cui si trova l'utente in quel dato istante. Il contesto è descritto in parte dalle proprietà fisiche dell'ambiente, come la posizione, l'attività svolta, il momento della giornata, il tempo meteorologico e la posizione semantica, ma per l'analisi dell'interrompibilità il contesto include anche lo stato interno del soggetto, come ad esempio il suo coinvolgimento in una specifica attività, l'ambiente sociale e le emozioni.

A monte di queste considerazioni, è necessario anche valutare la presenza di un cambiamento di contesto dal momento in cui l'utente riceve la notifica sul proprio dispositivo al momento in cui fornisce un feedback al sistema (ipotizzando vi sia una soglia massima di tempo accettabile in cui l'utente possa inoltrare la risposta). Tale fattore è rilevante per lo studio poiché il cambiamento di contesto può essere significativo per determinare lo stato di interrompibilità dell'utente. Ad esempio è possibile considerare tale scenario: l'utente si trova in chiesa quando riceve una notifica sul proprio dispositivo ma fornisce un feedback solamente quando si trova al di fuori dell'edificio. Pertanto il sistema da questa rilevazione impara che l'utente con maggior

probabilità sarà altamente interrompibile se è situato all'interno di un luogo di culto. Questo scenario esemplifica come il contesto possa essere funzionale al fine di definire lo stato di interrompibilità dell'utente. Per tale motivo all'interno della valutazione è necessario esaminare:

- il *contesto di notifica*, ovvero il contesto in cui si trova l'utente nel momento in cui viene ricevuta la notifica inviata dal sistema sul dispositivo utente;
- il *contesto di risposta*, ovvero il contesto in cui si trova l'utente nel momento in cui viene registrata una reazione dell'utente, ovvero un feedback che viene successivamente inviato al sistema.

Entrambi i contesti possono avere delle specializzazioni diverse a seconda se la sperimentazione viene eseguita in ambientazione indoor o outdoor e che permettono di considerare maggiori dettagli sulla caratterizzazione del contesto in cui si trova l'utente (il grado di dettaglio viene deciso in base all'interesse dello studio).

In relazione ai dati raccolti, è possibile ampliare lo studio integrando le metriche usate per misurare il successo dell'interruzione con l'analisi del contesto al fine di caratterizzare meglio il grado di interrompibilità dell'utente durante lo svolgimento di un'attività. In questo modo si identificano i legami tra:

- il contesto e la presenza della reazione dell'utente;
- tra il contesto e la tempestività della risposta;
- tra il contesto e il sentimento che un utente ha verso l'interruzione.

Pertanto, grazie a queste metriche di valutazione, è possibile assumere che l'analisi del contesto di notifica, integrata con le metriche usate per misurare il successo dell'interruzione, possa essere utile per predire i momenti opportuni in cui inoltrare una notifica poiché permette di ipotizzare se vi sarà la presenza di una reazione da parte dell'utente ed anche quanto tempo impiegherà per fornire una risposta.

Inoltre è importante evidenziare che in questo studio non si considera la concomitanza delle notifiche visualizzate sull'interfaccia poiché si suppone che il sistema debba riconoscere una singola attività svolta da uno specifico utente. Potrebbe invece verificarsi una concomitanza di notifiche nel caso in cui il wizard decida di raccoglierle all'interno di una coda, poiché accumulate da attività precedenti, e da inoltrare solo nel momento in cui l'utente non stia svolgendo alcuna attività. Inoltre è necessario specificare che in questo studio non verrà presa in considerazione la disponibilità dell'utente nell'utilizzo di sistemi volti al riconoscimento di attività umane e della conseguente interazione con l'interfaccia poiché si suppone che il soggetto sia interessato all'utilizzo di questi dispositivi.

2.4 Ciclo di vita di una query

Considerando quanto descritto nelle sezioni precedenti e gli obiettivi da raggiungere, è possibile stabilire quali sono le informazioni da tenere in considerazione per ogni query inviata dal sistema durante il suo “ciclo di vita” al fine di compiere la valutazione dello stato di accettabilità dell’interruzione provocata. Il “ciclo di vita” di una query è descritto nella **Figura 2** e comprende il seguente percorso:

1. formulazione query ed invio dal dispositivo wizard al dispositivo utente;
2. ricezione della query sul dispositivo utente;
3. formulazione della risposta ed invio al sistema (in questa sperimentazione la risposta verrà inviata al dispositivo wizard).

Le informazioni da tener in considerazione sono fornite dalla compilazione di alcuni campi, che verranno definiti di seguito, da parte del wizard durante ogni fase del ciclo di vita di una query. La compilazione di tali campi comincia prima che la notifica venga inoltrata sull’interfaccia tablet e comprendono:

- due attività di opzione tra cui l’utente può scegliere quando riceve la query sul proprio dispositivo;
- l’attività corretta (*ground truth*), siccome è importante verificare se l’attività selezionata dall’utente coincide con l’attività realmente svolta;
- il contesto di notifica, ovvero il contesto in cui si trova l’utente nel momento in cui il sistema decide di inoltrare la notifica (in questo caso è l’operatore a decidere il momento in cui inoltrarla).

Si ricorda che è necessaria la compilazione manuale di tali campi da parte del wizard poiché all’utente non viene fornito né un sistema di riconoscimento di attività né un sistema di rilevazione del contesto e pertanto, come precedentemente esposto all’interno della sezione 1.4, il wizard si occuperà di simulare sia il funzionamento dei sensori sui dispositivi che l’intelligenza del sistema al fine di non fondare la sperimentazione sulla casualità dell’effettivo errore di valutazione di un sistema funzionante.

Dopodiché, nella seconda fase del ciclo di vita della query, l’utente riceve sul proprio dispositivo la notifica e viene aggiunto ai campi iniziali il timestamp relativo al momento in cui la richiesta di feedback appare effettivamente sull’interfaccia. Infine l’utente compila la risposta e la inoltra al sistema (ovvero al dispositivo wizard). In quest’ultima fase del ciclo di vita della query, vengono aggiunti i seguenti campi:

- l’opzione selezionata dall’utente relativa all’attività svolta;

- il sentimento, ovvero la valutazione della reazione istantanea alla notifica da parte dell’utente;
- il timestamp relativo al momento in cui viene fornita la risposta e di conseguenza inviata al dispositivo wizard (questa informazione è fondamentale per calcolare nell’analisi successiva la latenza ovvero il tempo impiegato per fornire un feedback al sistema);
- il contesto di risposta, ovvero il contesto in cui si trova l’utente nel momento in cui inoltra la risposta al sistema e di conseguenza al dispositivo wizard (tale informazione è utile per verificare se è stato effettuato un cambiamento di contesto significativo per la valutazione dell’interrompibilità dell’utente).

È importante notare che gli ultimi 3 campi verranno compilati solamente se vi è l’effettiva presenza di una risposta da parte dell’utente, ovvero se il soggetto ha fornito un feedback relativo all’attività svolta. Viceversa, questi campi e la risposta dell’utente rimarranno vuoti. Inoltre è necessario ricordare che sia il contesto di notifica che il contesto di risposta comprendono non solo la posizione semantica in cui si trova l’utente ma anche la definizione.

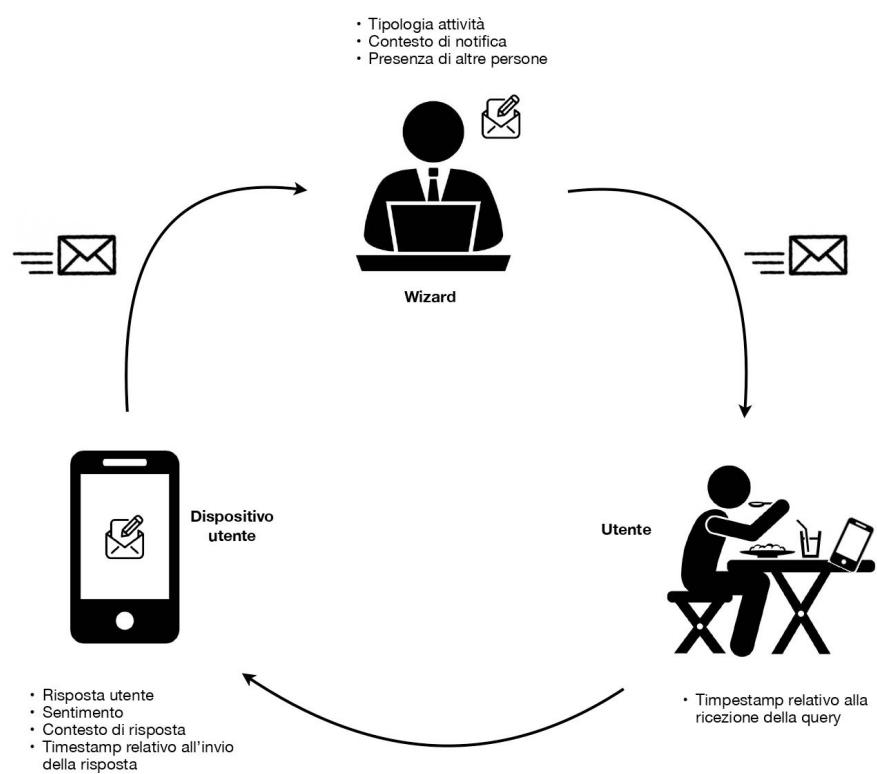


Figura 2: Schema riassuntivo della compilazione dei campi necessari per l'analisi finale durante il ciclo di vita di una query.

2.5 Definizione delle attività

Come anticipato nella sezione 1.4, le attività svolte dagli utenti vengono distinte in: attività di basso livello (attività caratterizzate solo da un movimento fisico e quindi più facilmente riconoscibili dal sistema) oppure attività di alto livello (attività caratterizzate da una concomitanza di movimenti fisici, pertanto più difficilmente riconoscibili dal sistema). Solitamente in letteratura, in base alla tipologia di sensori disponibili, è più comune monitorare le attività di basso livello in ambientazioni outdoor (anche se vengono eseguite molto anche in ambientazioni indoor) poiché più facilmente riconoscibili tramite l'utilizzo di sensori indossabili che monitorano movimenti fisici, mentre nelle ambientazioni indoor, grazie all'utilizzo di sensori ambientali, è più facile monitorare le attività di alto livello.

Si considera come possibile set di attività di basso livello svolte dagli utenti: *camminare, correre, stare in piedi, stare sdraiato, stare seduto, salire le scale, scendere le scale, salire con l'ascensore, scendere con l'ascensore, andare in bicicletta, guidare un mezzo di trasporto, stare seduti sui mezzi di trasporto, stare in piedi sui mezzi di trasporto.*

Per quanto riguarda, invece, le attività di alto livello si prendono in analisi: *spacchiare la tavola, cucinare, mangiare, preparare un pasto freddo, apparecchiare la tavola, prendere le medicine, lavare i piatti, piegare i vestiti, usare il computer, guardare la tv.*

Tali set di attività possono essere svolti in due scenari ovvero ambientazione indoor oppure ambientazione outdoor. È necessario ribadire (come descritto all'interno della sezione 1.2) che per lo scenario indoor, quale ad esempio una smart-home, si monitorano prettamente attività di alto livello e per quanto concerne l'informazione di contesto relativa alla posizione semantica si considera una granularità maggiore rispetto all'ambientazione outdoor. Infatti, a seconda del livello di dettaglio che si vuole raggiungere durante la sperimentazione, è possibile considerare ad esempio l'informazione relativa alla stanza in cui è situato l'utente durante la ricezione della notifica o l'invio di un feedback, oppure potrebbe interessare una granularità più fine della posizione, ad esempio si potrebbe definire se l'utente si trova davanti ad un lavandino o ad un elettrodomestico. Tale approccio è necessario per verificare se all'interno di uno scenario indoor esistono ambienti più significativi di altri per la valutazione dell'interrompibilità dell'utente. Per quanto riguarda lo scenario outdoor, invece, è più comune monitorare le attività di basso livello e si andrà a considerare come ulteriore informazione di contesto la condizione meteorologica.

Inoltre siccome è importante dare la possibilità all’utente di rispondere tramite diverse interfacce in entrambe le ambientazioni, si è deciso di utilizzare per questa sperimentazione l’interfaccia touch e l’interfaccia vocale. In questo modo si rende più agevole l’invio di un feedback da parte dell’utente anche se è fisicamente impegnato. Pertanto all’interno di questa sperimentazione le interfacce touch o vocali impiegate per comunicare con il sistema sono utilizzate su dispositivi tablet e smartwatch per l’ambientazione indoor mentre su dispositivi smartwatch e cuffie per l’ambientazione outdoor.

2.6 Modalità sperimentale

Prima di andare ad effettuare la sperimentazione, è necessario capire come istruire l’utente per svolgere l’esperimento. Pertanto è possibile considerare tre scenari:

- l’utente svolge le attività in piena autonomia, ovvero all’utente viene fornito un set abbastanza ampio di attività e potrà scegliere sia l’ordine di esecuzione che la tipologia di attività da svolgere in sequenza (siccome è un set abbastanza ampio è possibile prevedere che non verranno eseguite tutte le attività durante la sessione di sperimentazione);
- prima di iniziare lo studio il wizard comunica all’utente quali attività dovrà svolgere, ovvero verrà fornito all’utente un set di attività abbastanza piccolo che dovrà svolgere e sarà l’operatore a decidere se dovranno essere svolte con un preciso ordine oppure se la decisione è a carico dell’utente;
- il wizard comunica all’utente, in tempo reale e tramite cuffie, quali attività svolgere, in questo modo l’utente è completamente pilotato nell’esperimento e l’operatore è in grado di prestare maggior attenzione all’utente e non essere fuorviato dalla classificazione dell’attività che l’utente sta svolgendo (e pertanto ha maggior controllo sull’esperimento).

La prima modalità proposta risulta essere la più flessibile e realistica poiché l’utente non è pilotato dal wizard durante la sperimentazione. Siccome però all’interno di questo studio si vogliono inviare al dispositivo utente delle query di active learning in momenti specifici durante lo svolgimento delle attività, impostando dei dubbi specifici (ad esempio se l’utente sta svolgendo l’attività di “correre” è possibile inserire all’interno della query il dubbio tra le attività di “correre” e “camminare”), si vuole valutare qual è la reazione dell’utente se viene interrotto in tre diverse modalità: in mezzo allo svolgimento dell’attività, ad attività conclusa e quando si trova in posizione di riposo (ovvero riceverà una coda di notifiche relative alle attività svolte precedentemente). Tale decisione durante la sperimentazione deve essere presa dal

wizard in relazione a cosa sta facendo l’utente. Per questo motivo la prima modalità proposta risulta essere poco gestibile o problematica per il wizard all’interno di questa sperimentazione, siccome oltre a decidere quando inviare le notifiche deve essere in grado anche di interpretare le azioni dell’utente in tempo reale, poiché le attività non vengono definite prima della sperimentazione ma è l’utente stesso a decidere quale attività svolgere e in che ordine.

Per quanto riguarda la seconda modalità proposta permette di focalizzare la sperimentazione su un set di attività abbastanza piccolo in modo tale che l’utente sia in grado di ricordarlo e si possa condurre una valutazione più accurata per uno specifico scenario. Pertanto è anche vero che le attività eseguite durante la sperimentazione condotta in tale modalità, richiede un considerevole costo sia in termini di tempo e che di sforzo.

L’ultima modalità proposta prevede che la sperimentazione sia completamente pilotata. In questo modo sarà più semplice per il wizard focalizzarsi su come organizzare l’andamento dell’esperimento, ovvero capire quando inviare la query e come comportarsi, poiché è lui stesso a comunicare come deve comportarsi l’utente. Si riscontra però la stessa difficoltà di esecuzione della prima modalità proposta siccome l’utilizzo delle cuffie in ambientazione outdoor dovrebbe essere dedicato all’uso dell’interfaccia vocale per rispondere al sistema mentre in ambientazione indoor l’utente deve eventualmente essere in grado di sentire cosa comunica l’interfaccia tablet. Per tali motivi, nel caso in cui il wizard dovesse comunicare costantemente con l’utente, non sarebbe possibile utilizzare questa interfaccia per lo scopo dell’esperimento, oltre a far percepire all’utente un minor realismo poiché l’esperimento risulterebbe essere troppo pilotato.

Pertanto dopo aver dato una panoramica dei vantaggi e dei svantaggi di ogni modalità, la proposta più promettente per la tipologia di esperimento proposto all’interno di questo studio risulta essere la seconda modalità e per tale motivo verrà utilizzata durante la fase di sperimentazione.

Inoltre un’ulteriore differenza nell’esecuzione della sperimentazione in ambientazione outdoor o indoor è relativa all’osservazione dell’utente mentre svolge le attività effettuata dal wizard e dai suoi collaboratori. Per lo scenario indoor viene utilizzata una telecamera che permetterà di visualizzare a video i movimenti dell’utente in tempo reale, per lo scenario outdoor invece dovranno monitorare l’utente a distanza in maniera tale da non compromettere la sperimentazione.

2.7 Invio della query

Come precedentemente esposto, questo studio si occupa dell'osservazione delle attività svolte dagli utenti all'interno di una riproduzione di smart-home oppure in ambienti esterni, siccome è finalizzato ad analizzare il grado di interrompibilità dell'utente causato dalla ricezione di query di active learning (pertanto si vuole valutare il grado di accettabilità delle interruzioni provocate dalla ricezione di notifiche inviate dal sistema, durante lo svolgimento di un'attività). In tali riproduzioni il wizard deve essere in grado di analizzare il contesto in cui si trova l'utente e decidere come impostare e quando inviare le query al soggetto in maniera tale da capire cosa valutare considerando un'ampia varietà di scenari che potrebbero verificarsi nella realtà quotidiana di un soggetto. Per tale motivo è necessaria una progettazione accurata da parte del wizard per capire quali attività interrompere, in che modo interromperle (notifica singola oppure coda di notifica), quando inviare la notifica di active learning (durante l'attività, ad attività conclusa oppure quando l'utente è in situazione di riposo) ed anche valutare il contesto in cui si trova l'utente (se vi è la presenza di altre persone, se vi è interazione, etc). Ad esempio il wizard potrà determinare il numero di persone presenti nella stanza, quali oggetti o attività verranno manipolati, se l'attività che si sta svolgendo è di tipo sociale ed altre situazioni che possono essere associate ad un diverso grado di interrompibilità per poter effettuare una valutazione più approfondita. Pertanto l'invio di una query è stabilito principalmente valutando la tipologia di attività che il soggetto sta svolgendo, la distanza temporale dall'invio dell'ultima notifica, dal numero di query che si pongono all'utente e dal contesto semantico in cui si trova (luogo religioso, parco, mezzi pubblici, ristorante, etc).

Oltre a considerare lo scenario generale in cui si trova l'utente e definire come impostare la query, il wizard deve essere in grado di valutare anche quando inoltrare la richiesta di feedback al dispositivo utente. Questo è necessario al fine di valutare se lo svolgimento di attività specifiche richiede un maggior coinvolgimento da parte dell'utente rispetto ad altre ed in questo modo si è in grado di ampliare la valutazione dell'accettabilità del soggetto coinvolto. Ad esempio è possibile considerare se la notifica viene inviata:

- durante lo svolgimento dell'attività;
- ad attività conclusa;
- in momenti in cui l'utente è fermo (ad esempio in piedi o seduto), in tal caso l'utente potrebbe ricevere più richieste accumulate in una coda e potrebbe dover fornire un feedback riguardo ad un'attività svolta in un lasso di tempo abbastanza distante (pertanto il riscontro potrebbe non essere veritiero o esser assente).

La prima opzione risulta essere la più semplice da effettuare: il wizard osserva l’utente, valuta l’attività in svolgimento, formula la query scegliendo le opzioni tra cui il soggetto è tenuto a scegliere e invia la query al dispositivo utente. In queste circostanze l’utente viene interrotto mentre sta svolgendo l’attività e dovrà appositamente fermarsi e dedicare il suo tempo a fornire un feedback al sistema. Tale interruzione può delle volte non provocare un cospicuo disturbo se l’attività svolta non arreca un alto grado di coinvolgimento, come ad esempio guardare la tv oppure stare seduto sui mezzi di trasporto, ma molte altre attività richiedono un maggior coinvolgimento del soggetto e bisogna considerare inoltre il possibile coinvolgimento di altre persone durante la loro esecuzione. Ad esempio se il soggetto si trova coinvolto in una conversazione mentre sta svolgendo l’attività molto probabilmente non sarà disponibile a fornire un feedback al sistema, inoltre si vuole valutare se la sola presenza di altre persone possa influenzare il comportamento dell’utente, ad esempio se si trova sui mezzi di trasporto il soggetto preferisce rispondere quando sono presenti meno persone oppure rimane indifferente quanto è affollato il luogo in cui si trova.

La seconda opzione prevede che l’utente non venga interrotto mentre sta svolgendo un’attività ma piuttosto quando l’ha conclusa. Tale opzione potrebbe risultare vantaggiosa per l’utente, poiché l’attività non è in corso ma è stata appena conclusa e pertanto l’utente potrebbe essere più propenso a fornire un feedback al sistema, allo stesso tempo però tale approccio risulterebbe difficoltoso per il wizard siccome, ad esempio, deve essere in grado di riconoscere quando l’attività è stata effettivamente conclusa e di conseguenza formulare ed inoltrare la query prima che l’utente incominci una nuova attività.

Infine l’ultima opzione comprende che il wizard stabilisca quando l’utente si trova in posizione di riposo, ovvero quando l’utente sta svolgendo delle attività che non includono un particolare coinvolgimento fisico, ad esempio “stare seduto” o “stare in piedi”, ed in tale circostanza può ricevere una o più query. In questo caso il wizard accumula le notifiche relative alle attività svolte fino a quel momento dall’utente e decide di inoltrarle contemporaneamente sul dispositivo utente quando il soggetto risulta essere in un momento più idoneo per fornire dei feedback al sistema. Tale approccio risulta essere positivo poiché l’utente, essendo poco coinvolto fisicamente nell’attività che sta svolgendo, potrebbe essere più disponibile per fornire una risposta al sistema ma al contempo la ricezione di un numero troppo elevato di notifiche oppure la richiesta di feedback riguardante un’attività svolta precedentemente, e di cui l’utente non ricorda più i dettagli, potrebbero compromettere l’accuratezza della risposta fornita.

2.8 Definizione della tipologia di prompt e delle interfacce

Prima di avviare una sessione per la sperimentazione il wizard è tenuto a decidere la tipologia di notifica da inoltrare all'utente. Ciò significa che dovrà scegliere se inoltrare all'utente:

- una singola notifica;
- una coda di notifiche.

La prima opzione prevede che l'utente riceva le notifiche singolarmente sul proprio dispositivo e senza tener conto dello stato di svolgimento dell'attività. Pertanto le singole notifiche possono essere inviate in qualsiasi momento: durante lo svolgimento dell'attività, ad attività conclusa oppure quando l'utente è in stato di riposo. La seconda opzione, invece, è finalizzata ad accumulare le query generate e inoltrate dal wizard per mostrarle all'utente solamente quando è in stato di riposo. Questo approccio permette all'utente di non esser disturbato durante l'esecuzione di attività con un grado di coinvolgimento alto ma allo stesso tempo prevede l'accumulo di diverse richieste relative ad attività svolte precedentemente. Per questo motivo le query devono essere specifiche a seconda della tipologia di notifiche. Ad esempio quando il wizard inoltra una query all'interno di una coda di notifiche è necessario dare un contesto temporale all'utente in riferimento a quale attività è rivolta la query rispetto al momento in cui la query è effettivamente arrivata, inoltre tale riferimento deve essere facilmente comprensibile all'utente (come ad esempio “*cosa stavi facendo cinque minuti fa?*”). Il prompt quindi consisterà nella selezione di un'attività tra 2 possibili opzioni, scelte dal wizard all'invio della query, e di un'ulteriore nel caso in cui il sistema non abbia dato la possibilità di scegliere l'attività corretta. Quest'ultima può essere la selezione di un'ulteriore opzione, ad esempio con dicitura “nessuna tra queste”, oppure può essere data la possibilità all'utente di inserire manualmente o vocalmente l'attività in modo tale che possa essere il soggetto stesso a fornire ed aggiungere l'attività corretta. In questo studio è stato scelto il primo metodo ovvero aggiungere l'opzione “nessuna tra queste” poiché si suppone che l'inserimento manuale o vocale abbia un maggior impatto sull'interrompibilità dell'utente.

Le interfacce utilizzate dall'utente per fornire un feedback al sistema sono:

- interfaccia touch;
- interfaccia vocale, fondamentale per dare la possibilità all'utente di rispondere anche quando è impossibilitato a rispondere tramite interfaccia touch (ad esempio se sta svolgendo l'attività di “lavare i piatti”).

Entrambe le interfacce possono essere utilizzate sui seguenti dispositivi:

- smartphone;
- smartwatch;
- tablet;
- cuffie.

Il dispositivo smartphone è utile per fornire dei feedback sia tramite l’interfaccia touch che vocale in entrambe le ambientazioni ma risulta essere meno pratico da utilizzare mentre l’utente svolge un’attività soprattutto se questa richiede un impegno fisico. Anche il dispositivo smartwatch è utile per fornire dei feedback sia tramite l’interfaccia touch che vocale in entrambe le ambientazioni e risulta essere più vantaggioso rispetto allo smartphone poiché indossabile e quindi poco invasivo mentre l’utente svolge un’attività. Il dispositivo tablet, utile anch’esso per fornire dei feedback sia tramite l’interfaccia touch che vocale, è utilizzabile maggiormente in uno scenario indoor poichè risulta essere ingombrante e poco pratico mentre l’utente svolge un’attività. Infatti tale dispositivo solitamente viene posizionato in un punto della smart-home ed emette un suono all’arrivo di una nuova query di active learning. Le cuffie, utili per fornire dei feedback tramite l’interfaccia vocale, possono essere utilizzate sia in ambientazioni indoor che in ambientazioni outdoor. Per questa sperimentazione non risultano essere del tutto idonee per l’ambientazione indoor perché non permette di sentire le query di active learning.

2.9 Valutazione post-esperimento

Lo scopo finale di questo progetto è valutare quali sono, e come determinare, i momenti ottimali in cui il sistema possa interrompere l’attività che sta svolgendo l’utente senza che arrechi un eccessivo fastidio in modo tale da massimizzare il numero di feedback corretti (inerenti all’attività) forniti dall’utente. Per tale motivo a conclusione di questo studio è possibile andare ad analizzare qual è stata la user experience, oltre dai dati ricavati durante la sperimentazione, anche andando ad approfondire l’esperienza valutata direttamente dall’utente per comprendere come abbia vissuto questa interazione e quali accorgimenti apporterebbe al sistema per una migliore usabilità (per usabilità si intende l’efficacia, l’efficienza e la soddisfazione definita dall’utilizzo di un sistema).

Per valutare la user experience è possibile utilizzare due strumenti in particolare: questionari ed interviste. I questionari sono utilizzati maggiormente per ottenere un

riscontro quantitativo ovvero è possibile considerarlo un metodo per assicurarsi una grande quantità di dati per la semplicità e la rapidità con il quale si può fornire una risposta e soprattutto facile da analizzare se formato da domande a risposta chiusa. Le interviste invece permettono al soggetto che ha svolto l'esperimento di poter contestualizzare meglio agli operatori la propria esperienza nell'utilizzo del sistema.

A differenza dei questionari, la presenza del ricercatore allontana l'eventualità che il soggetto interpreti in maniera errata le domande e inoltre, nel caso di una risposta non attinente, il ricercatore può riformulare la domanda per questo i dati raccolti godono di maggiore validità. Se confrontato con il questionario, il metodo dell'intervista si rivela decisamente più costoso in termini di tempo e di sforzo perché richiede la presenza del ricercatore durante l'intera durata della sessione di interviste. La presenza stessa del ricercatore, inoltre, può rivelarsi dannosa nel caso il soggetto non si sentisse a proprio agio. Infatti, mentre nel caso dei questionari i dati raccolgono opinioni marcatamente orientate verso gli estremi della scala di valutazione, durante le interviste i soggetti tendono a fornire risposte che si attestano prevalentemente su valori medi, diminuendone la rappresentatività. Questa valutazione è necessaria quindi per approfondire l'analisi effettuata durante la sperimentazione e considerare come l'utente ha vissuto l'esperienza.

Sarebbe interessante anche andare a verificare se considerato un sistema in grado di imparare e affinarsi tramite questi feedback, e quindi tenesse in considerazione tutti i parametri descritti nelle sezioni precedenti, l'utente valutasse il sistema di migliore impatto ed usabilità ed inoltre se effettivamente una migliore user experience possa portare a diminuire gli errori forniti nei feedback dell'utente.

Capitolo 3

Caso d'uso: riconoscimento di attività in smart-home lab

Questo capitolo descrive la progettazione di un caso d'uso dell'applicazione in un contesto indoor, in particolare all'interno di una simulazione di casa domotica. Gli attori che vi partecipano sono principalmente il wizard, che monitora il soggetto durante lo svolgimento delle attività e si occupa di formulare ed inviare le query (oltre a fornire le informazioni necessarie per la valutazione), e l'utente, che svolge le attività durante la sessione di sperimentazione ed interagisce con il sistema. All'interno di questo studio gli attori principali vengono affiancati da attori secondari (ovvero collaboratori o volontari) che, oltre ad aiutare il wizard a monitorare l'utente durante lo svolgimento delle attività, permettono di simulare lo svolgimento di attività in contesti in cui vi è la presenza di altre persone oppure di interazione sociale al fine di ampliare i possibili scenari considerati per la valutazione finale.

Una volta spiegato quali sono gli attori, vengono descritti i dispositivi utilizzati per condurre la sperimentazione (ovvero dispositivo wizard, dispositivo utente e dispositivo di memorizzazione) e viene delineato il loro ruolo all'interno di questo studio, o meglio quali informazioni è necessario ricavare da ogni dispositivo al fine di condurre la valutazione riguardo lo stato di accettabilità dell'utente relativo alla ricezione di notifiche di active learning.

Viene inoltre definito com'è strutturato il setup sperimentale per lo studio di questo caso d'uso all'interno di una smart-home lab nel laboratorio EveryWare Lab e vengono illustrati gli ambienti di cui è composta. Il wizard sarà in grado di monitorare l'utente grazie all'utilizzo di una telecamera posta all'interno della simulazione di casa domotica, in questo modo si cerca di evitare una possibile compromissione della sperimentazione dovuto alla presenza fisica del wizard (o di un suo collaboratore) per tenere traccia dello svolgimento.

Infine vengono esposte le fasi di esecuzione dell'esperimento, definendo quali sono le

azioni che il wizard e l'utente devono compiere (ad esempio all'utente viene fornito il dispositivo per poter inviare dei feedback al sistema e il wizard deve comunicare il set di attività che il soggetto deve svolgere durante la sperimentazione), le tipologie di scenario escogitate per focalizzare la sperimentazione rispetto a contesti ben delineati e viene specificata la sequenza di azioni necessarie per l'acquisizione dei dati (ad esempio attivare la telecamera, attivare l'applicazione, etc).

3.1 Requisiti del sistema

Questo studio si è focalizzato su un caso d'uso specifico ovvero sullo sviluppo di un esperimento in ambientazione indoor. Tale scelta è motivata dal fatto che in lavori futuri in questo ambiente si faranno davvero esperimenti di questo tipo.

Gli attori principali di tale sperimentazione sono:

- *wizard*, colui che pilota l'esperimento e simula il funzionamento del sistema di riconoscimento di attività umane con active learning. In particolare, viene simulata la presenza di ambiguità nella classificazione dell'attività svolta dall'utente e pertanto è possibile utilizzare le tecniche di active learning per interrogare in modo interattivo l'utente per risolvere l'incertezza fittizia del sistema.
- *utente*, colui che userà effettivamente il sistema ovvero il soggetto che dovrà fornire dei feedback al sistema attraverso l'interazione con dei dispositivi durante lo svolgimento delle attività richieste inizialmente dal wizard. Si ricorda che l'utente è ignaro del fatto che il sistema non sia realmente funzionante, o meglio l'utente userà il sistema senza esser consci che le notifiche di active learning non siano generate in automatico dal sistema ma bensì dal wizard.

Vi sono anche degli attori secondari, denominati come *collaboratori* o *volontari* (come anticipato nella sezione 2.2), che affiancheranno gli utenti durante la sperimentazione principalmente al fine di aiutare il wizard a monitorare l'utente durante l'esperimento ma anche al fine di interagire con l'utente durante lo svolgimento di alcune attività per simulare, e pertanto valutare, diverse situazioni comprendenti la presenza di persone oppure l'interazione sociale durante l'utilizzo del sistema. In questo modo è possibile ampliare il caso di studio ad una maggiore varietà di scenari da considerare per la valutazione dello stato di accettabilità dell'utente durante l'utilizzo del sistema.

Entrambi gli attori principali (ovvero il wizard e l'utente) sono forniti di dispositivi che permettono di eseguire delle applicazioni. Tali applicazioni sono in grado di comunicare tra di loro affinché il wizard possa inviare delle query, per comporre delle notifiche di active learning, agli utenti e gli utenti possano fornire una risposta per

risolvere l'ambiguità fittizia del sistema. In aggiunta, tali dispositivi devono comunicare tramite una connessione wireless che permetta di scambiare le informazioni anche a lungo raggio, in questo modo non è necessario che il wizard si trovi in posizioni molto ravvicinate rispetto all'utente così che la sperimentazione risulti essere più realistica da parte dell'utente.

In specifico vengono utilizzati:

1. **dispositivo wizard.** Tale dispositivo deve essere in grado di eseguire un'applicazione (*applicazione wizard*) che permetta di:
 - impostare delle *informazioni di sessione*. Tali informazioni non verranno modificate durante lo svolgimento della sperimentazione in uno specifico scenario (per scenario si intende l'elenco di attività che l'utente deve svolgere durante una specifica sessione dell'esperimento comprendente anche le informazioni di contesto), pertanto per modificarle è necessario creare una nuova sessione di sperimentazione. Le informazioni di sessione comprendono l'inserimento dell'*ID utente* a cui inviare le query, la selezione della *tipologia di notifica* (notifica singola oppure una coda di notifiche) e l'inserimento del *time to live* (ovvero il wizard stabilisce entro quanto tempo l'utente può fornire il feedback).
 - impostare delle *informazioni di query* per ogni notifica inviata all'utente. Queste informazioni comprendono la selezione delle attività che andranno a comporre la query da inoltrare al dispositivo utente (ovvero il wizard deve poter scegliere due attività, che l'applicazione dovrà inserire nella query da inoltrare, a partire da una lista fornita dal sistema e comprendente tutte le attività considerate in questa sperimentazione), la selezione dell'attività corretta (*ground truth*) per verificare se coincide con il feedback fornito dall'utente, la definizione del *contesto di notifica* ovvero il contesto in cui si trova l'utente nel momento in cui il wizard decide di mandargli la notifica (specificando la posizione semantica, il momento della giornata e la stanza della smart-home in cui si trova) e la specifica relativa alla presenza di altre persone (indicando se c'è stata interazione sociale).
 - inviare la query generata, associata al *time to live* precedentemente stabilito, al dispositivo utente.

Inoltre il dispositivo wizard deve essere in grado di ricevere in tempo reale i feedback forniti dall'utente (associati ai relativi timestamp) per poter inserire le informazioni relative al *contesto di risposta* (ovvero il wizard è tenuto a fornire le informazioni relative alla posizione semantica, il momento della giornata e la stanza della smart-home in cui si trova l'utente al momento dell'invio della

risposta). Le informazioni relative al contesto (*contesto di notifica* e *contesto di risposta*) serviranno per verificare se coincidono oppure se vi è stato un cambiamento e risultano essere significative per tale valutazione. Infine il wizard può tenere traccia dello sviluppo della sperimentazione visualizzando lo storico delle sessioni sperimentali effettuate con le rispettive query, inviate dal wizard, e feedback, ricevuti dall'utente. In questo modo il wizard riuscirà ad avere sotto controllo lo stato di avanzamento della sperimentazione.

2. dispositivo utente. Tale dispositivo deve essere in grado di eseguire un'applicazione (*applicazione utente*) che permetta di:

- ricevere le query inviate dal dispositivo del wizard.
- memorizzare il *timestamp* t_1 in cui viene ricevuta la notifica sul dispositivo.
- notificare l'utente, o meglio l'utente può essere notificato tramite due tipologie di interfacce (interfaccia grafica e interfaccia vocale) ed entrambe devono essere in grado di: visualizzare (o leggere nel caso dell'interfaccia vocale) le query inviate dal dispositivo wizard, visualizzare (o leggere) le opzioni per fornire una risposta, permettere di fornire la valutazione del sentimento ed ottenere la risposta di entrambi i questionari.
- ottenere il *timestamp* t_2 nel momento in cui l'utente interagisce col dispositivo per selezionare l'attività.
- inviare la risposta al dispositivo di memorizzazione. La risposta verrà inviata nulla allo scadere del timer, impostato precedentemente dal wizard, altrimenti l'applicazione deve esser in grado di inviare la risposta fornita dall'utente con le *informazioni di risposta* le quali comprendono: ID dell'utente, l'attività selezionata dall'utente in risposta alla query ricevuta, valutazione del sentimento, timestamp t_1 del momento in cui la query viene notificata all'utente e timestamp t_2 del momento in cui viene fornita la risposta.

3. dispositivo di memorizzazione. Tale dispositivo deve essere in grado di memorizzare per ogni query le seguenti informazioni:

- *informazioni di sessione*, ottenute dal dispositivo wizard;
- *informazioni di query*, ottenute dal dispositivo wizard;
- *informazioni di risposta*, ottenute dal dispositivo utente.

Di seguito viene fornito uno schema riassuntivo (**Figura 3**) delle informazioni che ogni dispositivo deve essere in grado di fornire.

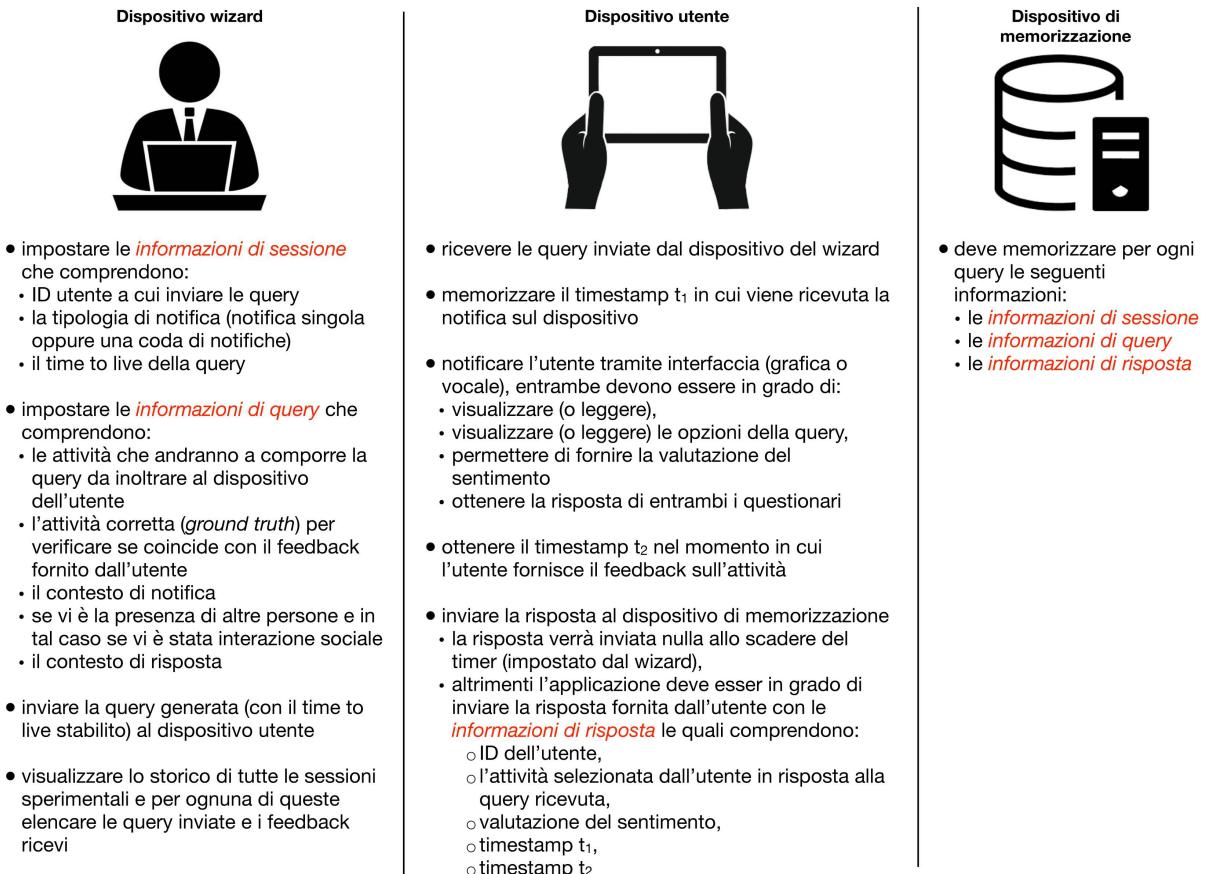


Figura 3: Schema riassuntivo delle funzionalità e delle informazioni relative ad ogni dispositivo utilizzato nella sperimentazione.

3.2 Setup sperimentale

Siccome questo studio è stato incentrato su un caso d'uso specifico relativo al contesto indoor, la sperimentazione sarà svolta in toto all'interno del laboratorio EveryWare Lab (**Figura 4**) appositamente organizzato al fine di simulare una smart-home.



Figura 4: Schema esplicativo della suddivisione degli ambienti all'interno del laboratorio EveryWare Lab.

Nella **Figura 4** viene illustrata la piantina della smart-home lab e pertanto è possibile osservare la disposizione degli ambienti per l'acquisizione dei dati. La smart-home lab è suddivisa in 6 aree a rappresentare le 6 diverse stanze della simulazione di casa domotica, in dettaglio:

- ingresso della smart-home;
- cucina utilizzata per preparare i pasti, lavare i piatti, prendere gli alimenti dal frigorifero o dai cassetti o armadi;

- sala da pranzo dove si può consumare il pasto;
- stanza delle medicine dove prendere le medicine;
- soggiorno in cui si può guardare la tv;
- ufficio in cui si può utilizzare il computer.

All'interno della figura inoltre viene specificata la presenza di una telecamera utilizzata esclusivamente per monitorare l'utente all'interno della smart-home durante la sperimentazione. Tale telecamera sarà collegata ad un laptop posto in una seconda stanza, adiacente alla smart-home, di modo che il wizard possa osservare l'utente svolgere le attività in tempo reale senza compromettere la sperimentazione ed anche capire quando ritiene opportuno inviare le query. È stato scelto questo approccio poiché con molta probabilità la presenza fisica del wizard durante la sperimentazione all'interno della smart-home avrebbe fatto percepire lo scenario meno realistico all'utente, compromettendo in questo modo la valutazione finale dell'esperimento.

3.3 Fasi di esecuzione dell'esperimento

Di seguito vengono descritte le fasi di esecuzione dell'esperimento svolto all'interno della simulazione di smart-home.

- L'utente viene fornito di un dispositivo tablet mentre il wizard ha a disposizione un laptop.
- Viene mostrato l'ambiente all'utente (ovvero la smart-home lab e come è organizzata), viene spiegato quali attività si possono svolgere e come (ad esempio come simulare l'attività di “lavare i piatti”).
- Successivamente, prima di svolgere l'esperimento, viene comunicato all'utente lo scenario dell'esperimento o meglio il wizard fornirà un set di attività non troppo ampio in modo tale da essere facilmente ricordato dall'utente senza un preciso ordine di esecuzione. Pertanto all'utente viene comunicato:
 - un set di attività da svolgere (senza un preciso ordine di esecuzione);
 - un tempo minimo di esecuzione per ogni attività.
- Durante lo svolgimento delle attività, il wizard monitora il soggetto e si occupa di formulare ed inviare le query al dispositivo utente. Come affermato nella sezione 2.7, il wizard deve essere in grado di capire cosa valutare durante la sperimentazione per decidere quando inoltrare le query all'utente. Pertanto

serve una progettazione accurata da parte del wizard per capire quali attività interrompere, quale tipologia di notifica usare, quando inviare le query, etc. Per ogni attività svolta, l’utente riceve una o più notifiche (nel caso in cui avvenga un cambiamento di contesto durante lo svolgimento della medesima attività).

- Le notifiche relative alle attività presenti nel set verranno inviate durante o alla fine dello svolgimento delle attività oppure vengono accumulate all’interno di una coda e verranno inoltrate all’applicazione utente solamente quando l’utente si trova in stato di “riposo”.
 - Se l’utente non fornisce i feedback al sistema, le notifiche vengono accumulate all’interno di una coda in attesa di ricevere una risposta fino al termine di un timer precedentemente impostato dal wizard. Allo scadere del timer la risposta viene memorizzata come nulla.
- All’interno del set di attività fornite all’utente, una percentuale verranno svolte con interazione sociale per simulare tale specifico contesto.
 - Dopo aver fornito una risposta alla notifica ricevuta, l’utente deve rendere noto al sistema la propria valutazione del *sentimento*, ovvero la valutazione della reazione istantanea alla query da parte dell’utente.
 - Terminato lo svolgimento del set di attività richieste, l’utente viene sottoposto ad una valutazione post esperimento attraverso un’intervista. Questa fase è utile perché permette agli operatori di capire più approfonditamente l’esperienza che ha avuto l’utente utilizzando il sistema. Siccome i quesiti richiedono un feedback di tipo qualitativo e pertanto non è possibile ricavarlo mediante l’utilizzo di scale di valutazione ma piuttosto con domande che necessitano risposte più articolate, risulta essere maggiormente indicato per questo scopo l’utilizzo di interviste rispetto ai questionari. Di seguito vengono indicate alcune possibili domande da porre all’utente:
 - quale dispositivo ha preferito usare per interagire col sistema e in quale contesto (siccome un medesimo utente potrebbe svolgere l’esperimento più volte con dispositivi e modalità di notifica diversa, in questo modo, dunque, è possibile verificare se il soggetto esprime una preferenza di utilizzo verso uno specifico dispositivo in un determinato contesto);
 - quale contesto è risultato di maggior disturbo per la ricezione di notifiche;
 - che tipo di sensazioni o suggerimenti ha rispetto all’uso di un sistema simile.

3.3.1 Definizione scenari ed attività

Al fine di condurre una sperimentazione più mirata per comprendere lo stato di accettabilità dell’utente causato dalla ricezione di notifiche di active learning, sono stati preventivamente stabiliti alcuni scenari pensati per essere svolti da un singolo utente. Ogni scenario richiede all’utente di simulare una sequenza di attività che verosimilmente vengono svolte nell’arco di una giornata all’interno di una smart-home.

Come esposto nella sezione 2.5, siccome in letteratura è più comune monitorare le attività di basso livello in ambientazioni outdoor poiché più facilmente riconoscibili tramite l’utilizzo di sensori indossabili che monitorano movimenti fisici, mentre nelle ambientazioni indoor, grazie all’utilizzo di sensori ambientali, è più facile monitorare le attività di alto livello, le attività considerate in questa sperimentazione sono prettamente attività di alto livello. Si ricorda che le attività di alto livello comprendono: *sparcchiare la tavola, cucinare, mangiare, preparare un pasto freddo, apparecchiare la tavola, prendere le medicine, lavare i piatti, usare il computer, guardare la tv.*

Affinché sia possibile effettuare una valutazione più accurata, si considerano tre scenari che delineano tre gradi di coinvolgimento differenti dell’utente durante lo svolgimento delle attività. Ognuno di questi scenari prevede l’esecuzione di tre attività da parte dell’utente:

1. *scenario statico*, formato da attività che non implicano un particolare coinvolgimento fisico da parte dell’utente e pertanto tale scenario dovrebbe indicare che durante lo svolgimento di queste attività l’utente è più propenso ad essere interrotto. Esso comprende le seguenti attività:
 - usare il computer, l’utente si siede sulla sedia dell’ufficio e simula di utilizzare un computer;
 - guardare la tv, l’utente accende il televisore e si siede su una sedia del salotto;
 - stare in piedi.
2. *scenario dinamico*, formato da attività che implicano un maggior coinvolgimento fisico da parte dell’utente e pertanto tale scenario dovrebbe indicare che durante lo svolgimento di queste attività l’utente è meno propenso ad essere interrotto. Esso comprende le seguenti attività:
 - prendere le medicine, l’utente estrae un medicinale dall’armadio delle medicine e lo assume eventualmente con un bicchiere riempito con l’acqua presa dal frigorifero;

- apparecchiare la tavola, l'utente stende la tovaglia sul tavolo, successivamente dispone il piatto, il bicchiere, le posate, il tovagliolo e una o più bevande presenti nel frigorifero;
 - preparare un pasto freddo, l'utente prende il pane, posto in dispensa, lo taglia a metà con un coltello preso dal cassetto delle posate e lo farcisce con degli alimenti estratti dal frigorifero.
3. *scenario sociale*. Tale scenario comprende la presenza di altre persone oppure un'interazione sociale in concomitanza con lo svolgimento di attività al fine di verificare se questi contesti possano influenzare lo stato di accettabilità dell'utente all'arrivo della notifica di active learning. Esso comprende le seguenti attività:
- stare in piedi;
 - mangiare, l'utente si siede su una sedia della sala da pranzo e simula di mangiare il piatto preventivamente cucinato;
 - lavare i piatti, l'utente simula di lavare i piatti a mano.

Il tempo di esecuzione stimato di un intero scenario è di circa 10 minuti.

Capitolo 4

Implementazione di un prototipo

In questo capitolo vengono mostrati i dettagli tecnici relativi all'implementazione del sistema proposto all'interno del capitolo 3.

Per l'implementazione dell'*interfaccia wizard* è stata sviluppata una WebApp in grado di creare una nuova sessione sperimentale, di formulare ed inviare delle query all'applicazione utente ed infine di visualizzare il riepilogo delle informazioni relative allo svolgimento dell'esperimento. Tale applicazione viene utilizzata all'interno di questo studio tramite un dispositivo laptop ma l'interfaccia è adattabile anche ad altre tipologie di dispositivo (ad esempio tramite dispositivi mobili in contesti outdoor).

Per l'implementazione dell'*interfaccia tablet*, invece, è stata estesa un'applicazione Android per tablet già esistente (utilizzata nel precedente lavoro di tesi di Andrea Cremonesi [7]) ed installata su un dispositivo tablet HTC Nexus 9. Tale applicazione è in grado di ricevere le query inviate dall'applicazione wizard e inoltrare la risposta fornita dall'utente (composta dall'attività selezionata e il sentimento).

Al fine di memorizzare le informazioni ricavate durante la sperimentazione per condurre la valutazione finale, è stato creato un database PostgreSQL. Questo database viene utilizzato dall'applicazione wizard in lettura (per visualizzare le attività che il sistema riconosce, al momento dell'invio di una nuova query, ed anche per la visualizzazione di riepilogo delle informazioni della sessione corrente o dell'ultima sessione o di tutte le sessioni) ed anche in scrittura (per la memorizzazione delle *informazioni di sessione*, *informazioni di query* e *informazioni di risposta* mostrate all'interno della sezione 3.1).

Infine nell'ultima sezione viene descritta la comunicazione, per l'invio di query e ricezione del feedback lato wizard e per la ricezione della query e l'invio del feedback lato utente, tramite l'utilizzo di socket (ovvero un'astrazione software progettata per la trasmissione e la ricezione di dati attraverso una rete).

4.1 Architettura

In **Figura 5** viene mostrato il flusso di comunicazione tra le applicazioni utilizzate e in particolare tra i moduli impiegati all'interno.

L'*applicazione wizard* è una WebApp realizzata nel seguente modo: la parte front-end è stata implementata attraverso i linguaggi HTML, CSS e Javascript, mentre la parte back-end è stata realizzata tramite il linguaggio PHP che si interfaccia con PostgreSQL per la lettura e scrittura su un database e con socket per permettere la comunicazione con l'applicazione tablet. L'*applicazione tablet*, invece, è un'applicazione Android per tablet che non è stata creata interamente per questa sperimentazione ma era già presente nello studio di Andrea Cremonesi [7]. In questo studio si è voluto estendere le funzionalità di tale applicazione con l'implementazione di una socket che potesse ricevere le query dall'applicazione wizard (e successivamente inviarle all'applicazione wizard) e di una nuova interfaccia formata da due Activity: *QueryAnswer*, *QueryFeeling*.

Il flusso di comunicazione tra le applicazioni inizia dall'interazione del wizard con l'interfaccia dell'*applicazione wizard* (le interfacce verranno illustrate e descritte con maggior dettaglio all'interno della sezione 4.2). L'interfaccia permette al wizard di interagire col sistema di modo che sia possibile iniziare una sessione, formulare ed inviare le query oppure visualizzare lo storico dei risultati ottenuti nella sessione corrente (o dell'ultima sessione se non è in corso di svolgimento) o nelle sessioni precedenti. Per poter effettuare queste operazioni però viene utilizzata la parte server della WebApp realizzata con PHP. Essa permette di:

- leggere e memorizzare le informazioni ottenute durante lo svolgimento della sperimentazione in un database PostgreSQL (la struttura del database è descritta all'interno della sezione 4.3);
- comunicare con l'applicazione tablet tramite socket (la comunicazione tramite socket verrà spiegata nella sezione 4.4), ovvero permette l'invio della query sul dispositivo utente e successivamente la ricezione della risposta sul dispositivo wizard.

Una volta inviata la query dal dispositivo wizard, l'applicazione tablet riceve la query tramite una connessione creata mediante socket. Tale socket una volta ricevuta la query, comunica le attività scelte dal wizard per impostare le opzioni della notifica di active learning e il time to live della query all'Activity *QueryAnswer*, la quale genera la schermata per richiedere il feedback dell'utente. Dopodiché quando l'utente seleziona l'attività viene mostrata l'Activity *QueryFeeling* per permettere all'utente di fornire la valutazione del sentimento relativa all'arrivo della notifica di active learning.

Successivamente la vista *QueryFeeling* comunica alla porzione di codice dedicata alla comunicazione tramite socket i risultati ottenuti dalle due Activity e la socket inoltre queste informazioni all'applicazione wizard tramite una connessione di rete via socket tra i due dispositivi, pertanto il requisito è che siano raggiungibili tra di loro nella stessa rete (o come avviene in questo studio, sottorete). Infine l'applicazione wizard, quando riceve il feedback dall'utente, permette al wizard di aggiungere le informazioni mancanti per la memorizzazione della risposta, passare dal server PHP per la memorizzazione dei dati sul database PostgreSQL.

Inoltre è importante specificare che in questa specifica implementazione del prototipo la parte relativa al server PHP e la parte relativa all'interfaccia vengono eseguite sulla stessa macchina ma in realtà nella versione finale del sistema la parte back-end verrà molto probabilmente sviluppata su una macchina diversa.

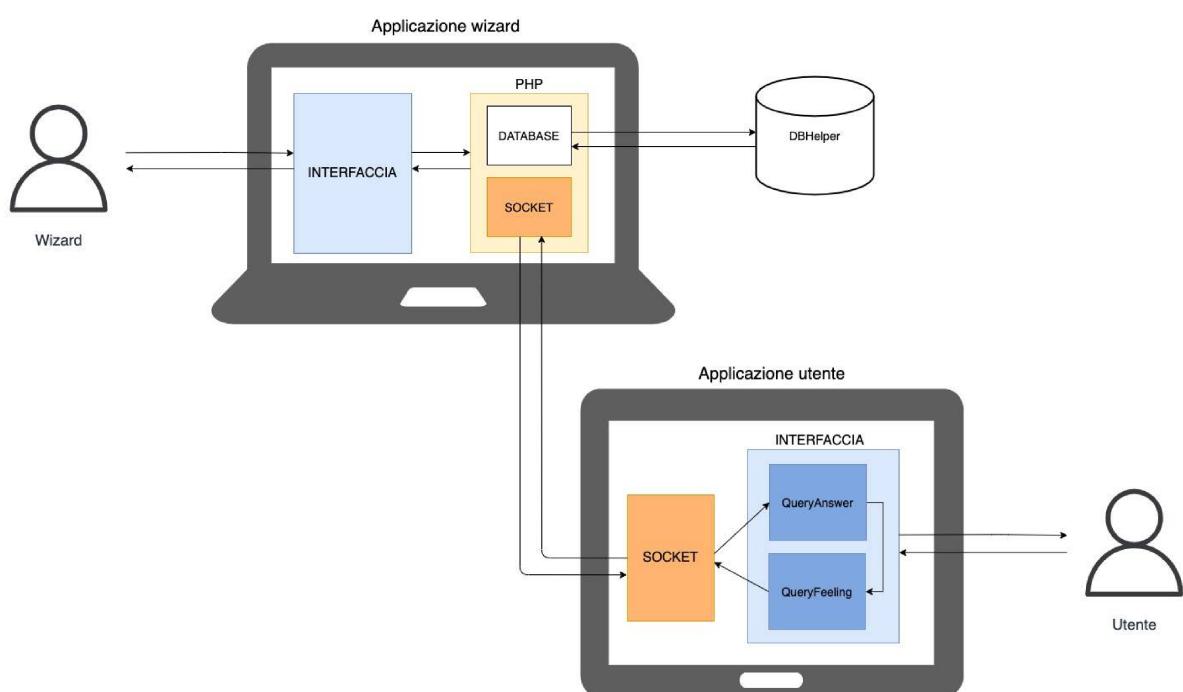


Figura 5: La figura mostra lo schema di navigazione tra le componenti del sistema.

4.2 Interfacce

All'interno di questa sezione vengono descritte ed illustrate le interfacce utilizzate dall'applicazione *wizard* e dall'applicazione *tablet*.

4.2.1 Interfaccia wizard

Per l'implementazione dell'interfaccia *wizard* è stata sviluppata una WebApp mediante l'utilizzo dei linguaggi HTML, CSS e JavaScript con un layout responsive al fine di poter essere visualizzata tramite diverse tipologie di dispositivi, in particolare in questo studio è stata utilizzata su un dispositivo laptop. All'avvio dell'applicazione viene mostrata la pagina relativa al login in cui il *wizard* deve autenticarsi attraverso il proprio ID, dopodiché può generare una nuova sessione impostando i parametri necessari. Successivamente il *wizard* può formulare ed inviare una nuova query all'applicazione *tablet* ed attendere la risposta, entrambi mediante l'utilizzo del server PHP che permette la comunicazione tramite socket. Una volta ricevuta la risposta è necessario che il *wizard* fornisca le informazioni relative al contesto di risposta per poter proseguire con la formulazione ed invio di successive query. Oltre a queste funzioni l'applicazione è in grado di far visualizzare lo storico della sessione attualmente in corso (oppure dell'ultima sessione) e delle sessioni precedenti.

Login e inizio sessione

All'avvio dell'applicazione *wizard* viene visualizzata la pagina di *Login* (**Figura 6**). In questa pagina il *wizard* deve autenticarsi, immettendo il proprio ID, per poter accedere ed utilizzare l'applicazione. L'autenticazione avviene lato server ovvero PHP e prende l'input della form e verifica se le credenziali immesse coincidono con quelle memorizzate all'interno del database.

Una volta effettuato l'accesso il *wizard* può decidere di visualizzare lo storico dell'ultima sessione o delle sessioni precedenti (**Figura 16**), oppure può decidere di generare una nuova sessione (**Figura 7**) o effettuare il logout. Per generare una nuova sessione è necessario che il *wizard* compili un form (**Figura 9**) selezionando la tipologia di notifica di active learning che si vuole inoltrare all'applicazione *tablet* ed indicando il *time to live* della query, ovvero l'intervallo di tempo entro il quale l'utente può fornire un feedback altrimenti viene inviato in automatico un feedback nullo. Dopo aver fornito e confermato le informazioni richieste, queste vengono memorizzate all'interno del database tramite un server PHP e successivamente viene visualizzata la *Home* dell'applicazione (**Figura 10**).

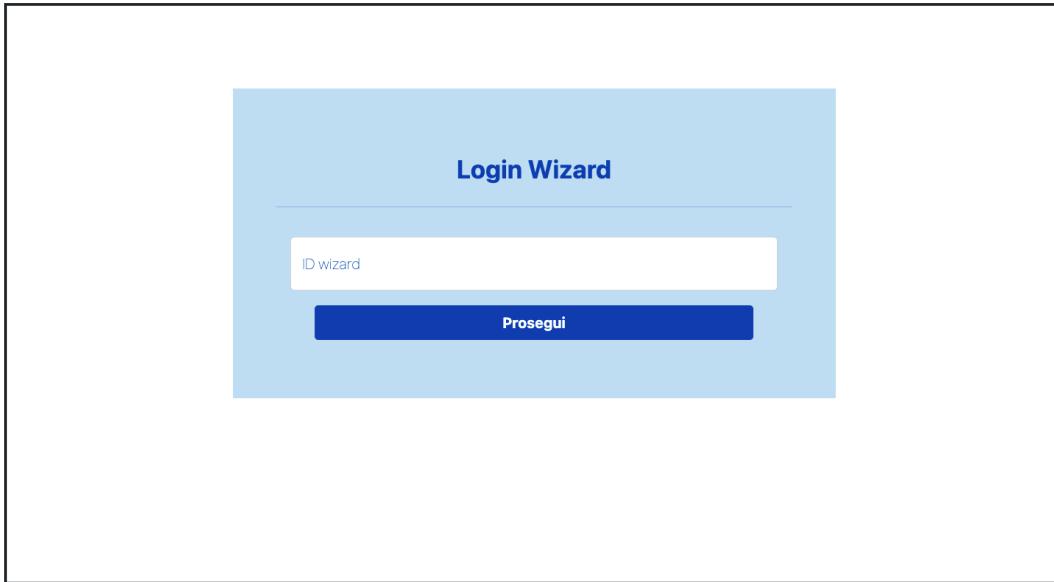


Figura 6: Schermata di *Login* in cui il wizard può inserire il proprio ID per accedere all'applicazione.

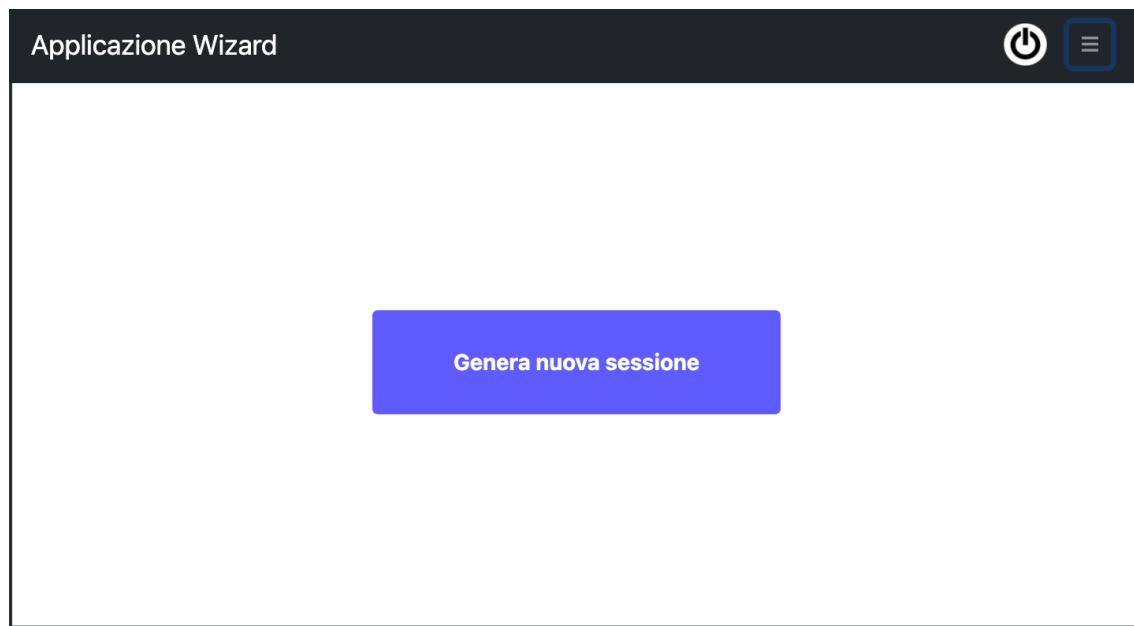


Figura 7: Schermata di *Home* in cui il wizard può decidere di generare una nuova sessione oppure visualizzare lo storico (dell'ultima sessione o di tutte le sessioni) oppure effettuare il logout (con il bottone in alto a destra).

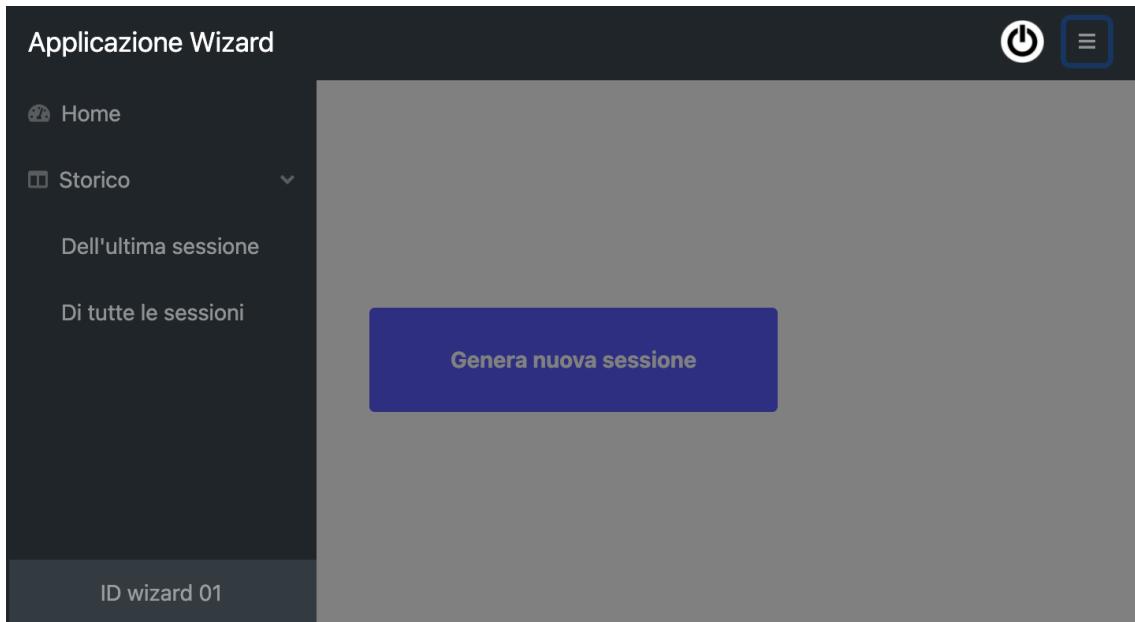


Figura 8: In questa schermata si mostra il menu apribile tramite il bottone in alto a destra.

A screenshot of the "Applicazione Wizard" configuration screen. It starts with a section titled "Tipologia di notifica:" containing a dropdown menu with three options: "notifica singola, durante l'attività" (selected), "durante lo svolgimento dell'attività" (radio button selected), and "ad attività conclusa". To the right of the dropdown is another radio button option "coda di notifiche". Below this is a "Time to live (sec.):" section with radio buttons for "30.00 sec.", "60.00 sec.", "120.00 sec.", and "altro" (selected). A text input field shows "0.00". At the bottom are three buttons: a green "Inizia nuova sessione", a blue "Pulisce form", and a red "Annulla e torna indietro".

Figura 9: Schermata mostrata per fornire le informazioni necessarie per creare una nuova sessione. I bottoni permettono la conferma dei dati inseriti, il ripristino della form oppure l'annullamento per tornare alla schermata in **Figura 7**.

Formulazione ed invio query

Iniziata una nuova sessione, il wizard può decidere di visualizzare il riepilogo delle informazioni di sessione (informazioni relative alla sessione corrente o di tutte le sessioni), di terminare la sessione, di effettuare il logout oppure può decidere di generare una nuova query (**Figura 10**). Per poter generare una nuova query è necessario che il wizard compili un form (**Figura 11**) fornendo le seguenti informazioni:

- due attività, che andranno a formare le opzioni all'interno della notifica di active learning e tra cui l'utente può scegliere per fornire una risposta;
- l'attività corretta (ground truth), necessaria per verificare se il feedback fornito dall'utente corrisponda l'attività realmente svolta o in svolgimento;
- la stanza della smart-home, siccome questo studio si focalizza su un caso d'uso all'interno di un contesto indoor;
- il momento della giornata in cui si invia la query;
- la presenza di altre persone, specificando, in caso affermativo, se vi è stata interazione.

Dopo aver confermato le informazioni inserite, vengono trasmesse al server PHP che si occupa di memorizzare queste informazioni all'interno del database e successivamente recuperate da uno script PHP dedicato alla comunicazione tramite socket per permettere l'invio della query. Tale script viene richiamato all'interno della *Home* mediante una chiamata asincrona AJAX. La maggior parte delle implementazioni AJAX usa l'API XMLHttpRequest, che include un elenco di richieste del server che possono essere richiamate all'interno del codice JavaScript (in particolare in questo prototipo viene utilizzata la libreria JavaScript jQuery). I dati vengono solitamente inviati al browser in formato XML, formato facilmente leggibile e manipolabile. Ciò che caratterizza AJAX è la leggera sfasatura temporanea tra la richiesta inviata dallo script al server web e la ricezione dei dati: avvenendo in momenti leggermente diversi, sono considerati asincroni. Inoltre gli script possono essere eseguiti lato client, consentendo di visualizzare immediatamente sulla pagina i dati ricevuti dal server, senza necessità di ricaricare la pagina stessa per visualizzarne i contenuti. In questo modo il wizard è in grado di interagire con l'interfaccia mentre AJAX si occupa di richiamare lo script PHP dedicato alla socket in background e pertanto la pagina HTML della *Home* si può aggiornare in modo automatico quando la risposta arriva dalla socket.

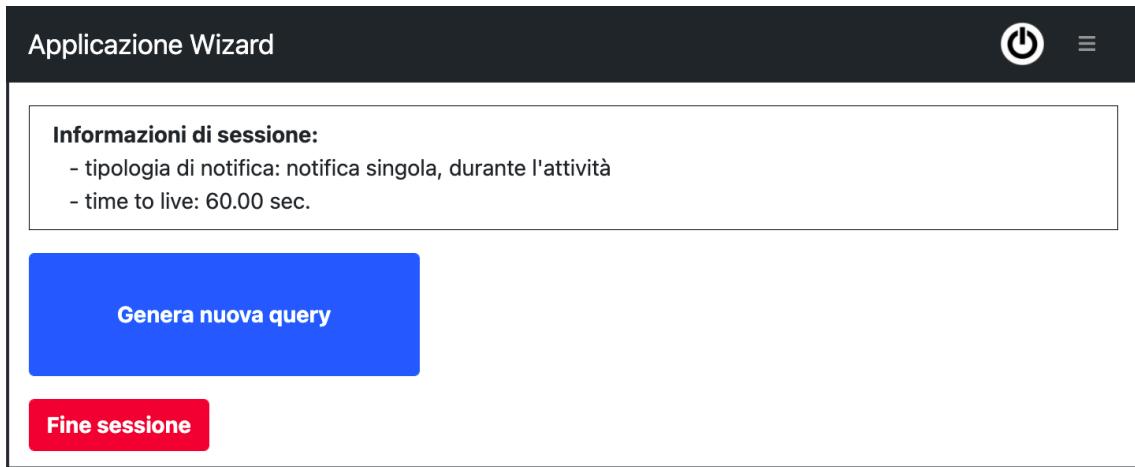


Figura 10: Visualizzazione della *Home* dopo aver generato la sessione. In questa pagina è possibile generare una nuova query, terminare la sessione, visualizzare lo storico (della sessione corrente o di tutte le sessioni) oppure effettuare il logout.

The screenshot shows the 'Applicazione Wizard' interface with the title 'Query 2' at the top right. It displays three sections: 'Attività di opzione:' with two dropdown menus ('opzione 1' and 'opzione 2'), 'Attività corretta:' with a dropdown menu ('ground truth'), and 'Contesto di notifica:' with a list of items and radio buttons. The 'Contesto di notifica:' section includes the following list and radio button groups:

- stanza smart-home
- momento della giornata
- presenza di altre persone

Radio button groups for context:

<input type="radio"/> cucina	<input type="radio"/> soggiorno	<input type="radio"/> sala da ufficio
<input type="radio"/> mattina	<input type="radio"/> pomeriggio	<input type="radio"/> sera
<input type="radio"/> sì, senza interazione	<input type="radio"/> sì, con interazione	<input type="radio"/> no

At the bottom are three buttons: 'Invia query' (green), 'Pulisci form' (blue), and 'Annulla e torna indietro' (red).

Figura 11: In figura viene mostrato il form relativo alle informazioni necessarie per formulare una nuova query. I bottoni visualizzati a fondo pagina permettono di confermare le informazioni inserite ed inviare la query, ripristinare il form oppure annullare e tornare alla schermata di *Home* (**Figura 10**)

Ricezione feedback utente

Per tenere traccia dello sviluppo della sperimentazione, all'interno della *Home* è possibile visualizzare le query che sono state precedentemente inviate ma che non hanno ancora ricevuto un feedback dall'utente e di cui non è ancora terminato il *time to live* (**Figura 12**). Questa operazione è possibile grazie all'utilizzo di un server PHP che interroga il database, verifica quali sono le query che non hanno ancora associata una risposta e permette di riepilogare le informazioni memorizzate. Il server PHP dedicato alla comunicazione socket si occupa sia di inviare le query al dispositivo utente che di ricevere il feedback dal dispositivo utente. Siccome prima di memorizzare all'interno del database la risposta dell'utente il wizard deve fornire le informazioni riguardanti il contesto di risposta, all'interno della schermata di *Home* è stata creata un'ulteriore chiamata AJAX che permette di visualizzare immediatamente sulla pagina il form per fornire tali informazioni (**Figura 13**) senza necessità di ricaricare la pagina stessa per visualizzarne i contenuti. Tale form è realizzato mediante l'utilizzo del server PHP che ne permette successivamente la memorizzazione sul database.

Visualizzato il form relativo al contesto di risposta, vengono disabilitati i buttoni dedicati alla creazione di una nuova query e alla cessione della sessione in corso poiché il wizard è tenuto a compilare e confermare le informazioni relative al contesto di risposta per poter effettuare la memorizzazione di tutte le informazioni relative alla risposta data dall'utente sul database mediante un server PHP.

È necessario specificare in aggiunta che quando viene visualizzato il form per il contesto di risposta, l'applicazione mostra delle opzioni già selezionate. Queste opzioni fanno riferimento alle informazioni relative al contesto di notifica, in questo modo si cerca di facilitare il wizard nella compilazione del form se il contesto al momento della risposta non è cambiato rispetto al contesto al momento della visualizzazione della notifica di active learning. Inoltre la compilazione del form per fornire le informazioni relative al contesto di risposta devono essere effettuata anche quando la risposta dell'utente non viene fornita poiché si vuole andare ad identificare il motivo per cui non è stato fornito un feedback e pertanto è necessario sapere in che tipo di contesto si trova allo scadere del timer e se è differente dal contesto di notifica.

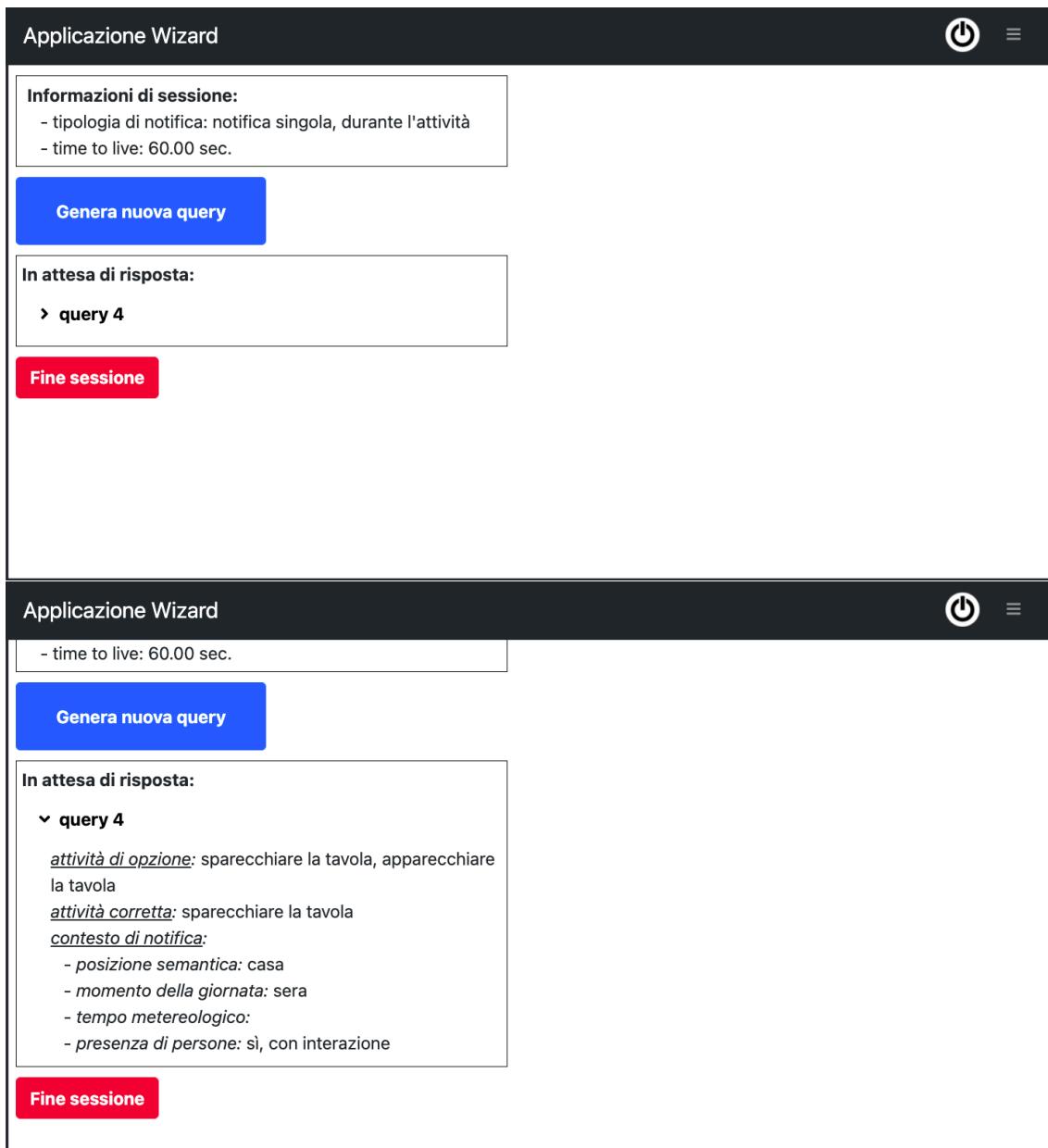


Figura 12: Schermata di *Home* con la visualizzazione delle query inviate che non hanno ancora ricevuto un feedback dall'utente e di cui non è ancora terminato il *time to live*.

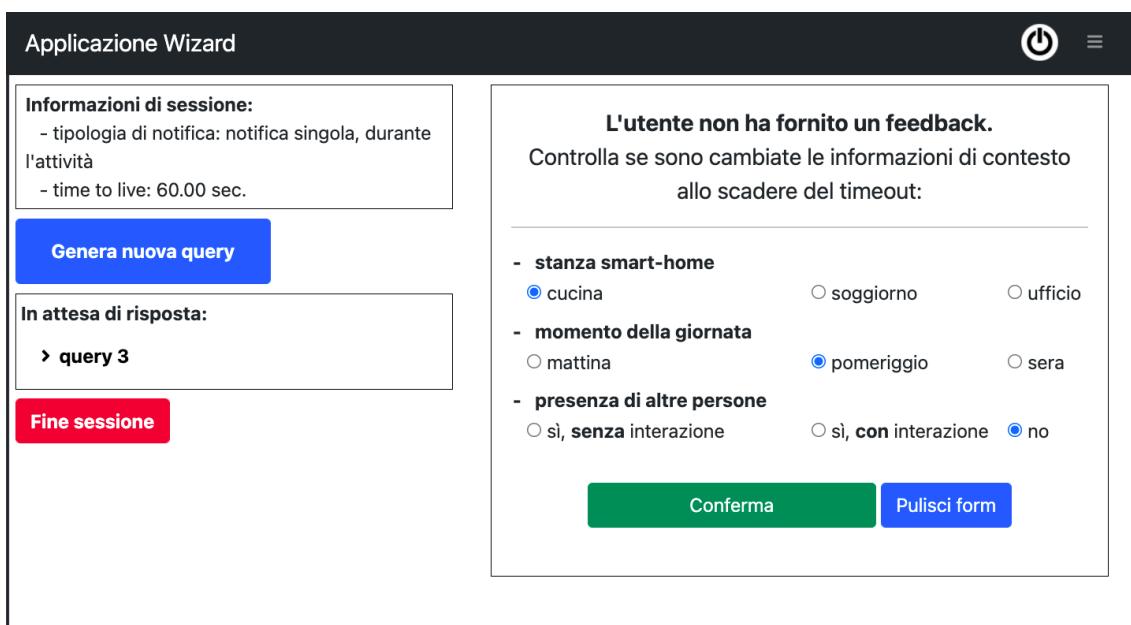


Figura 13: Visualizzazione della schermata di *Home* nel momento in cui l'applicazione riceve un feedback dall'utente. Prima di poter effettuare una nuova query o di terminare la sessione, il wizard deve fornire le informazioni relative al contesto di risposta per il feedback ricevuto in modo tale che l'applicazione possa memorizzare quanto fornito.

Storico

L’interfaccia wizard permette inoltre di verificare lo stato di avanzamento dell’perimento tramite delle pagine di riepilogo delle informazioni. Tali pagine permettono di visualizzare l’elenco di informazioni di sessione, di query e di risposta relativo all’ultima sessione (o sessione corrente nel caso sia attiva in quel momento) oppure di tutte le sessioni avviate durante la sperimentazione. Dal punto di vista tecnico, questa visualizzazione è permessa grazie alla creazione di un server PHP in grado di leggere sul database le informazioni e di associarle correttamente.



Figura 14: Schermata per la visualizzazione del riepilogo delle informazioni memorizzate all’interno della sessione in svolgimento.

Applicazione Wizard

Query 2

Informazioni di query

Attività di opzione: cucinare , mangiare
Attività corretta: cucinare
Contesto di notifica:

- **Posizione semantica:** casa
- **Momento della giornata:** pomeriggio
- **Stanza smart-home:** cucina
- **Presenza di persone:** no

Informazioni di risposta

Attività selezionata dall'utente: cucinare

Applicazione Wizard

Informazioni di risposta

Attività selezionata dall'utente: cucinare
Sentimento: mi ha infastidito molto
Timestamp: t1 = 15:00:00, t2 = 15:01:00
Latenza: Δt = t2 - t1 = 00:01:00.000
Contesto di risposta:

- **Posizione semantica:** casa
- **Momento della giornata:** pomeriggio
- **Stanza smart-home:** cucina
- **Presenza di persone:** no

Query 3

Figura 15: In questa schermata si mostra il dettaglio della Figura 14

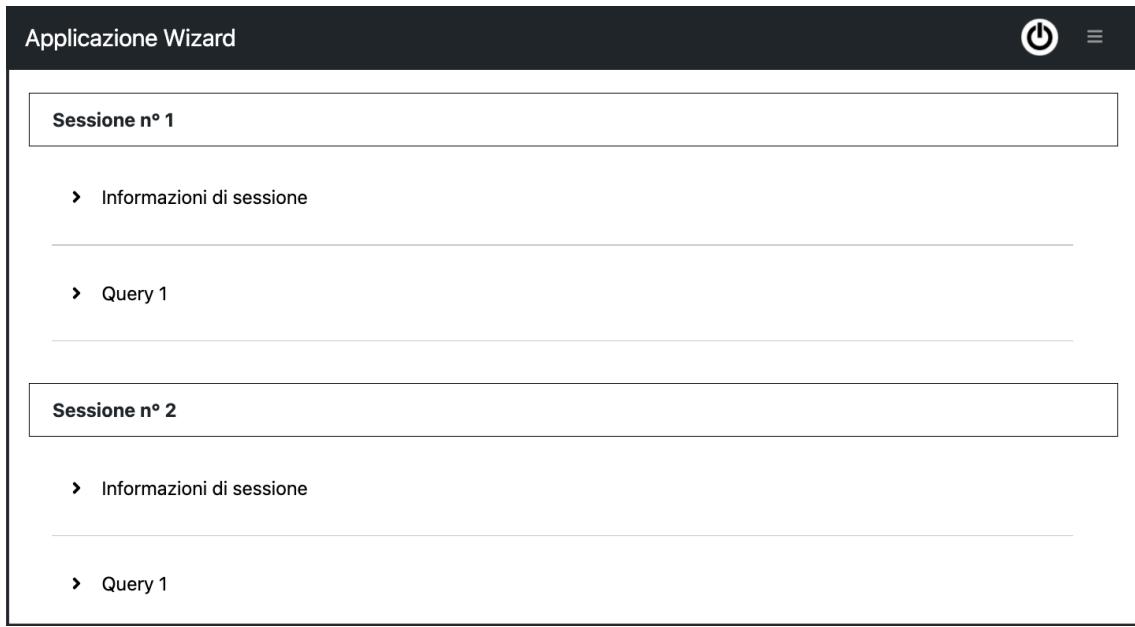


Figura 16: In questa figura viene mostrato il riepilogo delle informazioni relativo a tutte le sessioni effettuate durante la sperimentazione.

Fine sessione e logout

Infine l'interfaccia wizard dispone di due buttoni, visibili all'interno della **Figura 18** relativa alla *Home*: sotto all'opzione “Genera nuova query” vi è la possibilità di selezionare “Fine sessione” così da concludere la sessione in corso (**Figura 17**), mentre in alto sulla destra è possibile selezionare l'opzione di logout per uscire dall'applicazione e tornare alla fase di login (**Figura 6**). Entrambe le operazioni sono effettuate tramite dei moduli PHP che permettono di distruggere le variabili di sessione globali e permettere di utilizzare l'applicazione sul dispositivo ad un altro wizard.

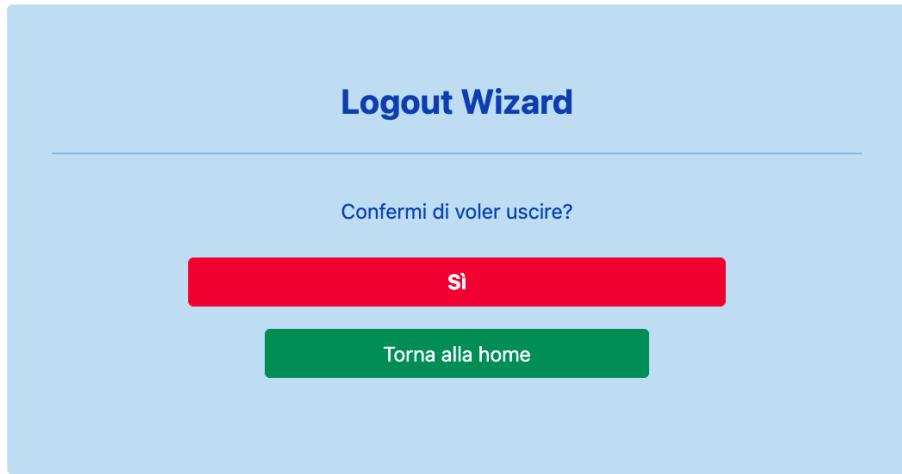


Figura 17: Schermata mostrata quando l’utente preme il bottone in alto a destra della *Home* (Figura 10) per effettuare il logout.

4.2.2 Interfaccia tablet

Per l’implementazione dell’interfaccia tablet è stata estesa un’applicazione Android per tablet già esistente nel lavoro di Andrea Cremonesi [7], installata su un tablet HTC Nexus 9. Il lavoro svolto nella precedente tesi si occupava dello sviluppo di un’interfaccia conversazionale smart-home in ambienti multi-inhabitant utilizzando tecniche di active learning. Tale interfaccia è stata realizzata affinché fosse possibile comunicare con gli utenti in modo intelligente, proponendo una soluzione ai problemi riguardanti la comunicazione in ambiente multiutente ovvero: la presenza di più soggetti all’interno di un ambiente smart (siccome rispetto ad un ambiente single-inhabitant è necessario determinare a quali utenti sono associati gli eventi legati ai sensori ambientali) e il riconoscimento dell’utente destinatario della notifica di active learning (pertanto il sistema deve essere in grado anche di comprendere quanto siano disponibili gli abitanti della casa ad interagire con l’interfaccia).

All’avvio dell’applicazione viene mostrata *HomeActivity* (**Figura 18**). Successivamente, quando l’applicazione tablet riceve una query sulla socket, viene aperta automaticamente l’Activity *QueryAnswer* che permette all’utente di fornire il feedback sull’attività corretta. Dopodiché, se l’utente fornisce una risposta alla notifica di active learning, l’applicazione visualizza automaticamente l’Activity *QueryFeeling* che permette al soggetto di fornire la valutazione sul sentimento. Una volta forniti entrambi i feedback l’applicazione torna sulla schermata iniziale *HomeActivity* ed attende l’arrivo di una successiva query.

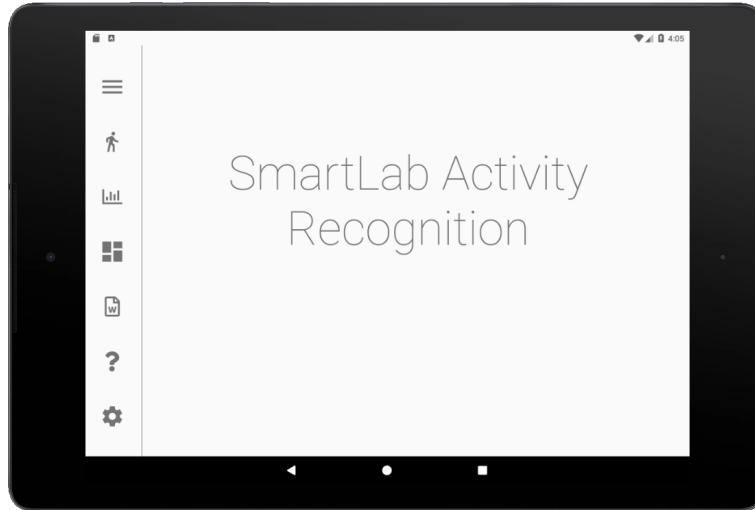


Figura 18: La figura mostra *HomeActivity* ovvero la schermata iniziale dell'applicazione.

QueryAnswer

Dal punto di vista tecnico *QueryAnswer* è un'Activity che viene inizializzata da un Intent all'interno del modulo dedicato alla socket poichè si occupa di mostrare all'utente la notifica di active learning all'arrivo della query sulla socket. Al momento dell'apertura automatica di *QueryAnswer* viene impostato un timer di lunghezza pari alla quantità di secondi comunicata attraverso il *time to live* fornito all'interno della query, inoltre la visualizzazione a schermo viene preannunciata da una notifica sonora, con lo scopo di catturare l'attenzione dell'utente. *QueryAnswer* permette all'utente di fornire un feedback scegliendo tra due possibili attività (fornite all'interno della query) oppure selezionando l'opzione “Nessuna tra queste” nel caso in cui non vi fosse l'attività corretta tra le attività proposte.

Il feedback dell'utente dev'essere fornito all'interno dell'intervallo di tempo prefissato dal *time to live* altrimenti, allo scadere di un intervallo di tempo prefissato, il tablet rimuove la vista *QueryAnswer*, invia tramite il modulo dedicato alla comunicazione tramite socket una risposta nulla all'applicazione wizard e visualizza la schermata *HomeActivity* aspettando l'arrivo di una nuova query.

QueryFeeling

Dal punto di vista tecnico *QueryFeeling* è un'Activity che viene generata da un Intent all'interno di *QueryAnswer* al momento in cui l'utente seleziona un'opzione tra quelle



Figura 19: La figura mostra l’Activity *QueryAnswer*.

proposte. Tale Activity permette all’utente di fornire la propria valutazione relativamente al grado di disturbo provocato dall’arrivo di una notifica di active learning e consente all’utente di scegliere tra tre gradi di disturbo con associati delle immagini simboliche per facilitare l’utente nella selezione della risposta corretta.

4.3 Memorizzazione dei dati

Le informazioni acquisite dall’applicazione wizard e dall’applicazione tablet (ovvero le *informazioni di sessione*, *informazioni di query* e le *informazioni di risposta*) vengono memorizzate all’interno di un database PostgreSQL. È importante sottolineare che è solamente l’applicazione wizard che comunica con il database poichè anche quando l’utente fornisce il feedback, la risposta viene inoltrata all’applicazione wizard che si occupa di aggiungere le informazioni relative al contesto di risposta e poi memorizzata all’interno del database.

Il database è formato dalle seguenti tabelle:

- **wizard:** Questa tabella viene utilizzata per memorizzare gli ID univoci dei wizard. In questo modo è possibile verificare le credenziali di accesso all’applicazione wizard durante il login.
- **inf_sessione:** Questa tabella viene utilizzata per memorizzare le informazioni relative ad ogni sessione. Una sessione viene identificata tramite l’ID della sessione associato all’ID del wizard ed è caratterizzata dalle informazioni relative



Figura 20: La figura mostra la vista *QueryFeeling*.

alla tipologia di notifica con cui si vuole inviare la query all’utente e al *time to live* in riferimento a quanti secondi la query rimane attiva in modo tale che l’utente possa fornire un feedback.

- **inf_query:** Questa tabella viene utilizzata per memorizzare le informazioni relative ad ogni query inviata dal wizard al dispositivo utente. Una query viene identificata tramite l’ID della sessione assieme al numero associato alla query ed è caratterizzata dalle informazioni relative alle due attività scelte dal wizard e poste come opzione all’interno della query, l’attività corretta che l’utente sta svolgendo (*ground truth*) e le informazioni relative al contesto di notifica la posizione semantica dove si trova l’utente al momento dell’invio della query (per motivi di tempistiche per lo sviluppo del sistema, all’interno di questo studio l’utente si trova sempre all’interno della simulazione di casa domotica), il momento della giornata in cui è stata inviata la query, la stanza in cui si trova l’utente quando la query viene inviata e la presenza di altre persone nel momento in cui viene inviata la query (specificando, in caso affermativo, se vi è stata anche un’interazione).
- **inf_risposta:** Questa tabella viene utilizzata per memorizzare la risposta fornita dall’utente e tutte le informazioni associate. Una risposta viene identificata dall’ID della sessione assieme al numero al numero associato alla query ed è caratterizzata dalle informazioni relative all’attività selezionata dall’utente, dal sentimento fornito (tale valore assieme all’attività scelta può essere nullo se l’utente non fornisce un feedback al sistema), il timestamp relativo al momento

in cui viene ricevuta la query sul dispositivo, il timestamp relativo a quando l'utente fornisce il feedback al sistema e le informazioni relative al contesto di risposta (quali posizione, daytime, meteo, persone)

- **attivita:** In questa tabella sono memorizzate tutte le attività riconoscibili dal sistema, inoltre ad ogni attività viene specificato se appartiene alla categoria di attività di alto livello oppure se appartiene alla categoria di attività di basso livello.

wizard	inf_sessione	inf_query	inf_risposta	attivita
id_wizard	id_sessione	id_sessione	id_sessione	attivita
	id_wizard	num_query	num_query	
	tipologia_notifica	attivita_opz1	attivita_scelta	
	ttl	attivita_opz2	sentimento	
		attivita_corretta	timestamp1	
		posizione	timestamp2	
		daytime	posizione	
		stanza	daytime	
		persone	stanza	
			persone	livello

Figura 21: In figura si mostra il riepilogo delle tabelle che compongono il database PostgreSQL. Per ogni tabella vengono elencati i campi di cui è composto indicando in grassetto le chiavi primarie.

4.4 Comunicazione tramite socket

La comunicazione tra l'applicazione wizard e l'applicazione tablet avviene tramite *socket* (con il termine *socket* si indica un'astrazione software progettata per la trasmissione e la ricezione di dati attraverso una rete). Tale modulo viene implementato all'interno dell'applicazione wizard tramite l'utilizzo di un server PHP che si occupa di:

- aprire la socket;
- connettersi al server;
- inviare il messaggio al server;
- leggere la risposta dal server;
- salvare la risposta in una variabile globale PHP;
- chiudere la socket.

All'interno invece dell'applicazione tablet, la comunicazione socket avviene all'interno del modulo *ServerSocketTask*.

In questo studio, siccome una comunicazione in rete è formata da processi client e processi server, è possibile identificare l'applicazione tablet come server mentre l'applicazione wizard come client. Questo perchè l'applicazione tablet apre una Server-Socket e si mette in ascolto su una specifica porta in attesa di connessioni mentre il wizard apre la connessione con uno specifico dispositivo, ovvero il tablet. Pertanto non è il dispositivo tablet a connettersi con l'applicazione wizard ma è l'applicazione wizard che deve conoscere esattamente l'indirizzo IP e la porta del tablet con cui deve comunicare. Per tale motivo un qualsiasi wizard si può connettere all'applicazione tablet ma il tablet con cui comunica è unico. Pertanto l'applicazione wizard utilizza in modo “hardcoded” un indirizzo IP e un numero di porta (ovvero si inserisce nel codice sorgente un indirizzo IP e un numero di porta statico) relativo allo specifico dispositivo per permettere la comunicazione. Inoltre, quando all'applicazione tablet arriva una connessione in entrata dall'applicazione wizard, riceve il messaggio, preleva le attività per impostare le opzioni all'interno della notifica di active learning, imposta il timer all'interno di *QueryAnswer* di lunghezza pari al valore del time to live e salva il numero della query a cui associare la risposta. Successivamente, dopo che l'utente ha fornito la risposta alla notifica di active learning e al sentimento relativo al grado di disturbo arrecato dall'arrivo della notifica o allo scadere del timer, viene utilizzato nuovamente questo modulo per scrivere sulla socket il messaggio di risposta in formato JSON ed inviarlo all'applicazione wizard. Infine, una volta finita

la comunicazione con l'applicazione wizard, il tablet chiude la socket relativa alla comunicazione di una specifica notifica di active learning e si rimette in ascolto di una comunicazione successiva sulla ServerSocket.

Capitolo 5

Conclusione e sviluppi futuri

Questo lavoro si è focalizzato sulla progettazione di un metodologia sperimentale, e la conseguente realizzazione di un prototipo, al fine di condurre una valutazione sullo stato di accettabilità degli utenti che utilizzano un sistema di activity recognition basato su active learning. La progettazione è stata realizzata tramite l'utilizzo del metodo Wizard of Oz che ha permesso di simulare il funzionamento di un sistema di activity recognition basato su active learning per poter ricavare delle informazioni in relazione al grado di interrompibilità dell'utente. Pertanto tale metodo consente una valutazione più accurata senza dover fondare la sperimentazione sulla casualità dell'effettivo errore di valutazione di un sistema funzionante. Tale progettazione ha delineato gli aspetti più significativi relativi al contesto in cui si trova l'utente al fine di stimare il grado di interrompibilità dell'utente.

La fase di implementazione ha incluso la realizzazione di una WebApp per permettere al wizard di creare ed inviare le query da mostrare come notifica di active learning sul dispositivo dell'utente, fornire le informazioni del contesto utili per la valutazione finale e tenere traccia dell'andamento della sperimentazione. Per permettere invece all'utente di fornire una risposta al sistema (in questo studio la risposta veniva inviata al wizard) riguardo all'attività in svolgimento ma anche il grado di disturbo che la notifica di active learning ha arrecato, è stata estesa un'applicazione Android per tablet utilizzata in un lavoro di tesi precedente [7].

È importante sottolineare però che durante l'implementazione del sistema sono emerse alcune debolezze.

Nell'attuale implementazione, il wizard deve memorizzare in modo "hardcoded" l'indirizzo IP e il numero di porta del client con il quale vuole comunicare. In questo studio viene utilizzato uno specifico tablet all'interno della smart-home lab. Pertanto il wizard non è in grado di scegliere a chi inviare le notifiche ma inoltra le query sempre al medesimo dispositivo il quale viene utilizzato da tutti gli utenti coinvolti nella sperimentazione. In realtà però la progettazione dell'esperimento è stata realizzata

per poter studiare diversi tipi di client e non rendere la sperimentazione vincolata ad uno specifico dispositivo.

Un altro problema è legato all'indirizzo IP poiché può anche cambiare in base a dove viene svolto l'esperimento. Di conseguenza la base per un lavoro futuro dovrà comprendere l'estensione del sistema in modo tale che il wizard e il client non debbano essere a conoscenza dei reciproci indirizzi di rete ma, piuttosto, utilizzare un server che metta in comunicazione il wizard con diversi clients.

Inoltre, in fase di progettazione, si è ipotizzata una modalità di query per la quale il client accumula quelle ricevute e le mostra assieme all'interno di una coda (ad esempio in un momento appropriato). Tuttavia, il prototipo sviluppato all'interno di questo lavoro di tesi permette di gestire solamente una query alla volta. In lavori futuri l'applicazione wizard e l'applicazione tablet dovranno essere estese per supportare questa funzionalità.

In questo lavoro di tesi è stato possibile valutare solo l'utilizzo di un'interfaccia tablet di tipo touch. In lavori futuri sarà necessario estendere il sistema su diversi client al fine di valutare, come precedentemente esposto, come diversi client impattano sull'accettabilità dell'interruzione provocata da un sistema di activity recognition basato su active learning.

Un altro miglioramento da effettuare sul prototipo è l'utilizzo dell'interfaccia vocale sia per la sintesi vocale della notifica di active learning arrivata sul dispositivo utente, che per il riconoscimento vocale. In questo modo l'utente è in grado di fornire un feedback al sistema, sia in ambientazioni outdoor che indoor, anche quando è impossibilitato ad utilizzare l'interfaccia touch (ad esempio se l'attività svolta include "lavare i piatti" oppure "guidare").

Un'ulteriore possibile miglioramento riguarda l'estensione del funzionamento del client anche su dispositivi mobili, come ad esempio su un dispositivo smartwatch.

Durante questo lavoro di tesi (per motivi di tempo) si è riusciti solo a progettare la metodologia sperimentale, capire i requisiti di uno specifico caso d'uso e realizzare un prototipo. Pertanto il prototipo non è ancora pronto per effettuare una valutazione vera e propria relativa all'accettabilità di sistemi di activity recognition basati su active learning (visti anche i problemi riportati prima) e per questo è necessario un ulteriore lavoro implementativo prima di realizzare una sperimentazione completa.

Bibliografia

- [1] J. Pansiot, D. Stoyanov, D. McIlwraith, B. Lo, G. Yang, “*Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems*”, In: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, Springer, RWTH Aachen Univ, Aachen, Germany, 2007, pp. 208–212.
- [2] X. Wang, D. Rosenblum, Y. Wang, “*Context-aware mobile music recommendation for daily activities*”, In: Proceedings of the ACM International Conference on Multimedia, ACM, New York, NY, USA, 2012, pp. 99–108.
- [3] Arpan Deyasi, Swapan Bhattacharyya, Pampa Debnath, Angsuman Sarkar, “*Computational Intelligence in Digital Pedagogy*”, vol. 197, pp. 1, 2021.
- [4] Takuya Maekawa, Shinji Watanabe, “*Unsupervised Activity Recognition with User’s Physical Characteristics Data*”, Wearable Computers (ISWC) 2011 15th Annual International Symposium on, pp. 89-96, 2011.
- [5] Hande Alemdar, T.L.M. van Kasteren, Cem Ersoy, “*Activity recognition with Hidden Markov models using active learning*”, Signal Processing and Communications Applications (SIU) 2011 IEEE 19th Conference on, pp. 1161-1164, 2011.
- [6] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook and Z. Yu, “*Sensor-Based Activity Recognition*”, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 790-808, Nov. 2012.
- [7] Andrea Cremonesi, “*Studio di un’interfaccia conversazionale per ambienti smart-home multiutente*”, 2019.
- [8] Claudio Bettini, Gabriele Civitarese, Riccardo Presotto, “*CAVIAR: Context-driven Active and Incremental Activity Recognition*”, Knowledge-Based Systems, Volume 196, 2020.

- [9] Nattaya Mairittha, Tittaya Mairittha, Paula Lago, and Sozo Inoue, *CrowdAct: Achieving High-Quality Crowdsourced Datasets in Mobile Activity Recognition*, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol, Article 50, p. 32, March 2021.
- [10] Carmelo Ardito, Paolo Buono, Maria Francesca Costabile, Rosa Lanzilotti, Antonio Piccinno, “*A tool for Wizard of Oz studies of multimodal mobile systems*”, Human System Interactions 2009 (HSI ’09), 2nd Conference on, pp. 344-347, 2009.
- [11] Hiromitsu Awano, Tetsuya Ogata, Shun Nishide, Toru Takahashi, Kazunori Komatani, Hiroshi G. Okuno, “*Human-robot cooperation in arrangement of objects using confidence measure of neuro-dynamical system*”, Systems Man and Cybernetics (SMC) 2010 IEEE International Conference on, pp. 2533-2538, 2010.
- [12] Rahat Iqbal, Kashif Iqbal, Nazaraf Shah, Jawed Siddiqui, Anne James, “*Development of Context-Aware Systems to Support Human-Human Communication and Collaboration*”, Information Technology: New Generations 2008. ITNG 2008. Fifth International Conference on, pp. 206-211, 2008.
- [13] Michael Schreiber, Michaela Kauer, Ralph Bruder, “*Conduct by wire - maneuver catalog for semi-autonomous vehicle guidance*”, Intelligent Vehicles Symposium 2009 IEEE, pp. 1279-1284, 2009.
- [14] Yang Li, Jason I. Hong, James A. Landay, “*Design Challenges and Principles for Wizard of Oz Testing of Location-Enhanced Applications*”, Pervasive Computing IEEE, vol. 6, no. 2, pp. 70-75, 2007.
- [15] Yoosoo Oh, Albrecht Schmidt, Woontack Woo, “*Designing Developing and Evaluating Context-Aware Systems*”, Multimedia and Ubiquitous Engineering 2007. MUE ’07. International Conference on, pp. 1158-1163, 2007.
- [16] Zahra Shakeri Hossein Abad, Shane D.V. Sims, Abdullah Cheema, Montasir B. Nasir, Payal Harisinghani, “*Learn More Pay Less! Lessons Learned from Applying the Wizard-of-Oz Technique for Exploring Mobile App Requirements*”, Requirements Engineering Conference Workshops (REW) 2017 IEEE 25th International, pp. 132-138, 2017.
- [17] S. Dow, B. MacIntyre, J. Lee, C. Oezbek, J. D. Bolter and M. Gandy, “*Wizard of Oz support throughout an iterative design process*”, in IEEE Pervasive Computing, vol. 4, no. 4, pp. 18-26, Oct.-Dec. 2005, doi: 10.1109/MPRV.2005.93.

- [18] Veljko Pejovic and Mirco Musolesi, “*InterruptMe: designing intelligent prompting mechanisms for pervasive applications*”, In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp ’14), Association for Computing Machinery, New York, NY, USA, 897–908, 2014.
- [19] Y. Miyata, D. A. Norman, Psychological Issues in Support of Multiple Activities, “*User Centered System Design: New Perspectives on Human-Computer Interaction*”, pages 265–284, 1986.