

**Seminario di cultura digitale**

# **Scenari End-User-Development per la Domotica**

**Luca Corcella**  
**Matricola: 527815**

## **Introduzione.**

In questi ultimi anni abbiamo assistito all'uso del computer in un numero crescente di contesti dinamici. Il numero e la varietà di dispositivi sono in aumento. Gli scenari di utilizzo possono variare dalla gestione, dall'ingegneria, dalla progettazione, dall'istruzione, dalla ricerca, dalle vendite, dall'amministrazione o da altri settori. Gli utenti condividono esigenze comuni per supportare le loro attività che possono variare rapidamente. I software che vengono usati per diversi scopi sono complessi e hanno esigenze diverse.

D'altro canto i cicli di sviluppo dei software sono lenti e la mancanza di conoscenze di applicazione del dominio da parte degli sviluppatori, rappresentano delle limitazioni per affrontare le esigenze di diversi utenti. L'end user development (EUD) può contribuire ad attenuare queste lacune. Per EUD si intende un insieme di metodi, tecniche e strumenti che consentono all'utente finale non esperto di programmazione, di modificare o estendere il software artefatto. I principali approcci EUD sono in genere, identificabili in quelle piattaforme in grado di adattarsi al mutevole contesto d'uso. Spesso queste tecniche vengono usate di supporto nel comporre e personalizzare elementi di base di un'applicazione.

Tali elementi di base sono rappresentati attraverso metafore intuitive, come il puzzle in cui gli elementi di base corrispondono ai pezzi da comporre o rappresentazioni iconiche dei dati in cui le icone corrispondono agli elementi di base. L'internet of things e gli ambienti intelligenti sono diventati fonte di attrazione per industrie e studiosi di ricerca in diversi campi, dall'intelligenza artificiale alla sicurezza e l'analisi di dati. La ricerca su nuovi paradigmi della Human Computer Interaction si concentra, in particolare, per rendere sempre più agevole la configurazione e l'uso di tali ambienti 'intelligenti' da parte di utenti non esperti di informatica. I sensori, che rendono le nostre abitazioni sempre più smart, stanno avendo un incremento esponenziale; l'utente che si interfaccia con queste tecnologie ha la necessità di modificare il comportamento a seconda delle proprie esigenze. L'uso di tool end user development consente di personalizzare il controllo dei

sensori domotici in maniera intuitiva e senza richiedere competenze informatiche particolari.

In commercio sono state sviluppate diverse applicazioni che consentono di creare funzioni dipendenti dal contesto ad utenti non professionisti. Le applicazioni di rilievo sono: Tasker, Atooma e Locale, che forniscono soluzioni diverse a seconda delle esigenze. Queste prevedono tre diverse soluzioni per supportare specifiche applicazioni dipendenti dal contesto secondo: lo stato, la condizione e il modello di azione evento. In particolare, un evento è un qualcosa che accade in un determinato momento, eventi elementari che si verificano nell'applicazione interattiva o nel contesto di utilizzo, o in una composizione di tali eventi. Una condizione è un vincolo specifico che deve essere soddisfatto, si può fare riferimento a qualcosa accaduto prima o qualche stato o condizione. Un'azione è la descrizione di come l'applicazione interattiva dovrebbe cambiare al fine di eseguire l'adattamento richiesto.

Una 'Smart Home' può essere vista come un ecosistema di computer, storage di dati, sensori, attuatori. La combinazione di questi apre la strada ad un alto potenziale da sfruttare, rendere la casa intelligente grazie alla combinazione del software giusto e hardware adeguato per soddisfare le esigenze degli abitanti. Nei diversi app-store vi sono svariate applicazioni che rispondono alle singole necessità degli utenti, ma questo grande mondo di applicazioni può essere difficile da gestire nel contesto domestico; in primo luogo i componenti tecnici di uno smartphone sono ben definiti, mentre quelli delle case intelligenti sono molto varie e imprevedibili, pertanto, le applicazioni per la domotica devono affrontare molte fonti di incertezza. Secondo, le applicazioni che si possono visualizzare sullo schermo possono essere una al massimo due, mentre gli scenari tipici delle case intelligenti prevedono un gran numero di servizi in esecuzione in parallelo con interfacce utente distribuite attraverso le risorse di interazione disponibili dentro casa. Lo scopo è quello di creare applicazioni mobile customizzabili per i propri ambienti domotici, avvalendoci dei

tool di End users development e utilizzando una filosofia di sviluppo che deriva dai vari principi della EUD.

## **La nascita dell'End User Development.**

EUD è nato da una confluenza di innovazioni incorporate nelle macchine note come "microcomputer" (un termine eventualmente sostituito con "personal computer"). Innanzitutto, queste macchine erano abbastanza economiche tanto da rendere possibile agli utenti di modificare ("su misura") le impostazioni del software e dell'elaboratore senza impattare l'ambiente di calcolo di altri utenti. In secondo luogo, i microcomputer avevano un hardware sufficiente potente, in modo che gli utenti potevano compilare (o interpretare) il nuovo codice in linguaggi di programmazione. Questo, a sua volta, ha fornito le infrastrutture per gli utenti finali creando nuove applicazioni. In terzo luogo, i personal computer hanno incluso le nuove caratteristiche innovative come il mouse e schede grafiche potenti, che hanno accelerato i progressi come le interfacce utente (GUI) e la manipolazione diretta; questi progressi, inoltre, hanno aperto la possibilità di creare nuovi strumenti di programmazione specificamente progettati per soddisfare le esigenze degli utenti.

## **L'approccio EUD per la domotica.**

Un uso dei tool EUD all'interno di un ambiente domotico è un approccio del tipo "do it yourself", che fornisce agli utenti gli strumenti adeguati per costruire la loro casa in base alle loro esigenze, invece di fare affidamento su sviluppatori che dovrebbero accrescere l'ambiente domotico prevedendo future modifiche. EUD lascia spazio anche alla creatività per personalizzare i propri dispositivi personali. Pertanto, è ragionevole pensare che gli utenti facciano lo stesso per la propria casa.

Questo approccio è adatto anche per affrontare un ambiente eterogeneo in cui ogni abitante possiede un dispositivo diverso configurabile a seconda delle proprie necessità e tenendo conto anche dei sensori domotici integrabili in un futuro. Inoltre, l'EUD può essere usato per creare da parte di un amministratore delle restrizioni, per esempio limitare l'accesso a particolari servizi della casa a determinate persone, o vietare la possibilità di modificare le funzionalità del programma a terzi. Le applicazioni commerciali per la domotica e i tool di end user development possono essere complementari, utilizzando app già esistenti si possono creare blocchi o combinazioni di funzionalità tramite i tool EUD. La possibilità di modificare e combinare devices, apps e servizi è il fulcro dell'EUD. Il passo successivo è quello di creare applicazioni che incorporano i tool EUD per produrre sistemi digitali all in one.

## **Il futuro della domotica.**

Come le case ordinarie si evolvono nel tempo con la ristrutturazione o l'acquisizione di nuovi oggetti, le case intelligenti saranno costruite in modo incrementale (scalabile), sia in termini hardware che software.

La scalabilità dei componenti domotici comporta una serie di problemi per l'utente finale: in primo luogo, l'installazione di nuovi dispositivi o servizi e l'eventuale sostituzione o l'eventuale modifica devono essere riprogrammati dagli utenti.

Poi, dispositivi, applicazioni e servizi possono essere interconnessi attraverso lo sviluppo di programmi da parte dell'utente finale, che a sua volta include la programmazione, debug, test e la condivisione e il recupero di dati locali o da luoghi pubblici.

L'utilizzo della domotica, oltre che in ambito residenziale, può essere estesa anche al terziario. In questo caso si parla di «building automation». Questo significa dar vita ad ambienti di lavoro dove, oltre agli impianti di illuminazione, riscaldamento e

sicurezza, anche le reti informatiche e di comunicazione sono integrate in un unico sistema, che ne semplifica e ottimizza il controllo e la gestione.

## **Installazione dei dispositivi.**

L'installazione di un componente domotico (vale a dire un dispositivo, un servizio o una nova app), dovrebbe essere completamente “plug and play”, per non dire completamente “put and play” soprattutto quando si tratta di dispositivi wireless quali EnOcean<sup>1</sup> o simili. In particolare, non dovrebbe esserci la necessità di configurare le caratteristiche hardware dei nuovi componenti. Un dispositivo dovrebbe integrarsi semplicemente settando un nome definito dall'utente, in modo, da rendere il componente domotico riconoscibile dagli utenti e dagli altri sistemi (ad esempio una casa con diversi televisori o diverse lampadine con settaggi personalizzati per ogni stanza dell'abitazione). Tuttavia, denominare ogni componente domotico della casa solleva diverse domande: in genere, in una casa multi-utente bisognerebbe autorizzare più nomi per lo stesso componente? Questo aiuterebbe ogni singolo abitante ad associare un nome? O si creerebbero solo incomprensioni tra gli abitanti? Oltre all'associazione di nomi, gli utenti dovrebbero essere in grado di associare qualsiasi meta-dato che reputino essere utile per la loro vita quotidiana, come l'utilizzo per il quale si è acquistato il dispositivo domotico.

L'associazione di metadati e nomi aiuta il sistema ad identificare un componente tecnicamente e quindi generare l'adeguata interfaccia utente (UI). Come possibile soluzione, l'identificazione può essere eseguita con uno smartphone o un tablet mediante l'uso di tecnologie quali: codice a barre, NFC, RFID, IBacon.

---

<sup>1</sup> enOcean è un'azienda che si occupa dello sviluppo di materiale elettrico per la casa con connettività wireless.

L'installazione di un nuovo componente non implica un sistema finito, ma bisogna prevedere eventuali cambi di nome dei componenti, spostamenti o rimozioni dei componenti.

Questo tipo di modifiche potrebbero avere forti impatti sugli impianti in esecuzione, quindi le smart homes dovrebbero essere in grado di fare queste dipendenze in maniera trasparente e dinamico.

## **Designazione.**

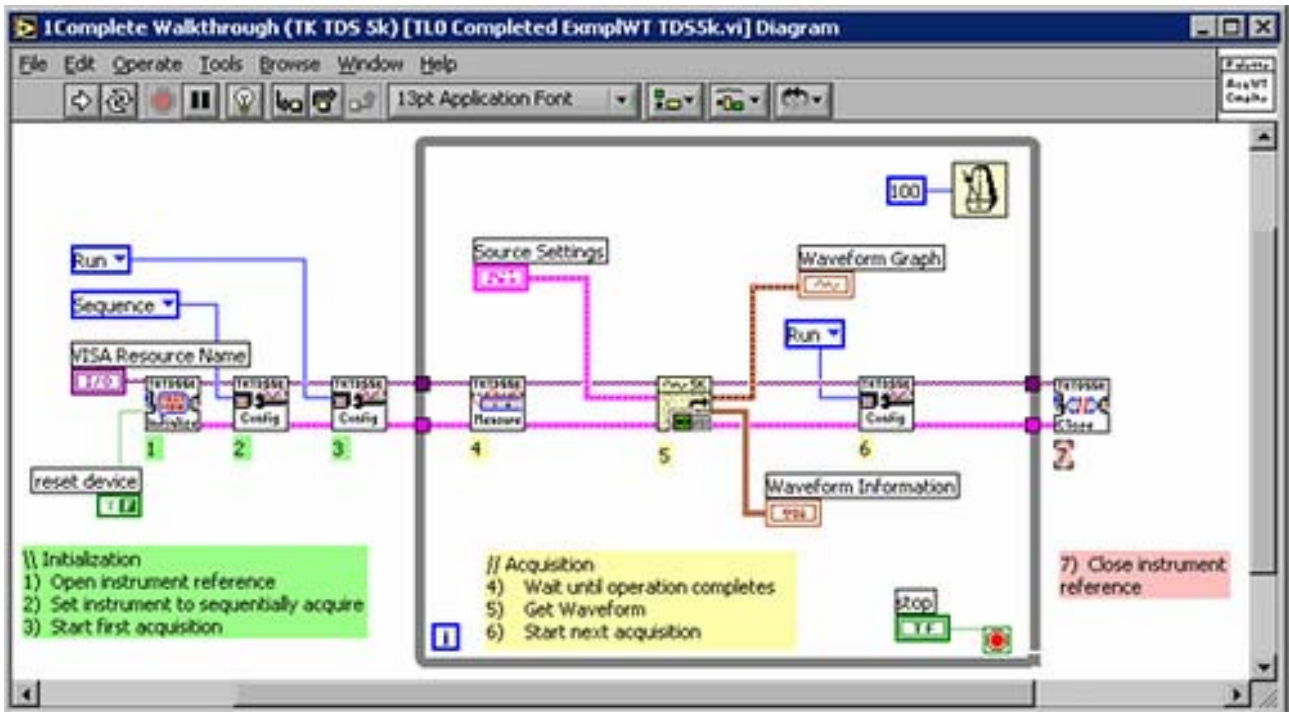
Una volta installato il componente, deve essere designato in modo che gli utenti possano controllarlo fin da subito (come di solito si controllano le TV con il telecomando). La designazione può essere diretta o indiretta. Quella diretta può essere ottenuta allo stesso modo come nella fase di installazione, associando in maniera diretta il componente (se fisico) o, altrimenti, associando una sua rappresentazione se il componente è digitale.

La designazione indiretta si riferisce ai paradigmi dei linguaggi di programmazione., utilizzando i nomi e i metadati associati ai componenti categorizzandoli con dei selettori, così facendo si renderà semplice l'assegnazione e la configurazione dei componenti domotici.

## **Controllo.**

Un componente designabile può essere controllato, la verifica avviene tramite l'interazione con le interfacce utente generate successivamente all'installazione dei componenti e, quindi, dopo aver creato il nostro ambiente domotico. Quando parliamo di interfacce per la domotica, esse non possono prescindere da determinate caratteristiche, oltre ad essere facilmente usabili e intuitive, dovranno essere anche

molto dinamiche e modulabili. Nello sviluppo di esse, quindi ,potremmo considerare indispensabile, l'utilizzo di tool End User Development. Strumenti visuali (figura 1 e 2)



(figura 1)



(figura 2)

Questi tool con interfaccia grafica modulabile ( detta anche a “puzzle”) consentono di create interfacce e logiche di controllo, senza avere competenze di



programmazione. Possiamo, quindi, sviluppare applicazioni in grado di controllare gli oggetti fisici o i sensori domotici, tramite uno smartphone e fornire anche un framework per consentirne la modifica.

## **Linguaggi e strumenti per lo sviluppo di programmi.**

Come abbiamo accennato, lo sviluppo di applicazioni per la domotica vede l'utilizzo delle tecniche di End User Development; questo implica la progettazione di nuovi linguaggi di programmazione e di ambienti di sviluppo ben congeniati. Vari test hanno dimostrato che gli utenti si trovano a loro agio con i paradigmi basati su regole di Evento – Condizione – Azione (ECA) . Tuttavia, la programmazione ECA sembra risultare difficile quando vi è un vasto numero di regole; al fine di facilitare il compito dell'approccio ECA, bisognerebbe proporre dei mezzi per raggruppare alcune regole. La struttura proposta sopra, che prevede l'installazione e la designazione dei componenti, potrebbe aiutare il raggruppamento delle regole di ECA in modo rilevante. Inoltre, vi è anche la necessità di utilizzare astrazioni ad alto livello.

Il linguaggio di programmazione deve quindi sostenere questo tipo di approccio e permettere agli utenti di esprimere azioni invece di procedure. Tali implementazioni saranno possibili mediante l'interfaccia di un qualsiasi dispositivo; utilizzando i paradigmi dei linguaggi visuali possiamo sviluppare sistemi con semplici descrizioni (ad esempio per una lampadina la parola che descriverà l'azione per accenderla sarà "on").

Il sistema EUD dovrebbe anche essere in grado di riconoscere nuove situazioni in base alle nuove necessità della casa. Infatti, è molto difficile che un utente possa essere capace di abbinare nuovi sensori installati a determinate situazioni casalinghe, tale compito risulterebbe molto complicato da parte di un utente non esperto di informatica; per questo il sistema EUD deve imparare da ciò che osserva e proporre

nuove situazioni agli utenti, in tal modo si renderebbe il sistema facilmente scalabile sia orizzontalmente che verticalmente.

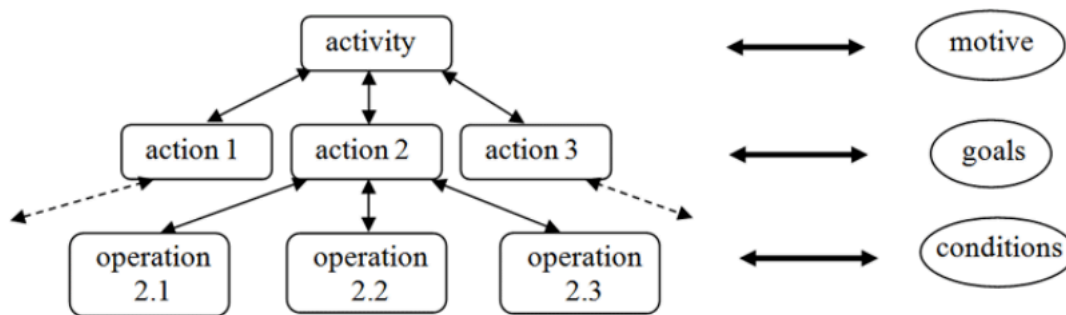
Questo approccio però solleva importanti sfide in termini di visualizzazione, ovvero come rendere la visualizzazione di un nuovo sistema che descriva “situazioni”, per renderlo più dinamico possibile? In questo ambito la Human Computer Interaction ci suggerisce diversi metodi da utilizzare, in primo luogo come per qualsiasi linguaggio (non solo di programmazione), vi è un compromesso tra espressività e semplicità. Ad esempio, i meccanismi offerti dovrebbero proporre astrazioni di regole o gruppi di regole? Dovrebbe essere possibile definire variabili e parametri? Bisognerebbe affidarci a metafore per la realizzazione di interfacce dei programmi o rappresentare l’astrazioni delle regole ? La HCI suggerisce che un linguaggio visuale dovrebbe essere abbastanza espressivo per poter definire azioni specifiche o situazioni (come quelle che si creano in un ambiente domotico). Si potrebbe analizzare uno scenario in cui in una casa abitata da più persone vi è il bisogno di limitare l’accesso ad alcuni dispositivi da parte di determinati utenti. Per far ciò il linguaggio del programma EUD dovrebbe essere abbastanza espressivo per essere in grado di definire politiche interne, ovvero specificare diritti di accesso ai servizi e dispositivi, ma anche di definire la risoluzione di conflitti, come ad esempio se determinate televisioni della casa vengono impostate con una restrizione e, quindi, gli abitanti minorenni non vi possano accedere, come dovrebbe comportarsi il sistema in queste situazioni?.

La politica, quindi, non deve essere codificata dal sistema, ma definita dagli utenti, per motivi di coerenza essa dovrebbe essere espressa in maniera analoga ad altri programmi. Analizzando queste problematiche, possiamo descrivere lo stato dell’arte attuale delle smart home; la visione di fondo della casa intelligente è come se essa fosse un grande computer con le sue periferiche che funzionano in maniera autonoma, ma la ricerca ha mosso la visione delle smart homes su altri scenari, suggerendo di considerare la domotica come dei servizi distribuiti che soddisfino le

esigenze delle persone, e quindi un sistema o un framework EUD è la via da seguire per implementare programmi o app per la domotica.

## L'organizzazione delle attività e approcci innovativi.

Secondo i ricercatori, le attività umane sono organizzate in livelli gerarchici (figura 3). Lo strato superiore è l'attività stessa che è orientata verso un determinato bisogno. Le azioni sono processi consci dirette ad obiettivi che devono essere intrapresi per soddisfare l'oggetto, gli obiettivi poi devono essere scomposti in sotto obiettivi. Le azioni vengono attuate mediante delle operazioni, le operazioni sono processi di routine che svolgono le azioni predefinite.



(figura 3)

Questa struttura gerarchica ci fornisce le giuste indicazioni per la realizzazione di strumenti per la domotica basati su un modello EUD differente al modello di Condizione – Evento - Azione . In primo luogo, le attività sono guidate da motivi soggettivi degli utenti, è molto improbabile che le attività possano essere comprese dal sistema EUD. In secondo luogo, le operazioni possono essere totalmente o parzialmente svolte con un sistema EUD (ad esempio, gestione della musica o accensione delle luci al verificarsi di una determinata condizione). In terzo luogo, le

relazioni tra le azioni e le operazioni non devono essere viste come delle procedure rigorose che dovrebbero essere soddisfatte dal sistema; al contrario, sono molto flessibili, improvvisazioni ed evoluzioni degli ambienti EUD che dovrebbero consentire agli abitanti di adattare rapidamente le loro necessità alle azioni e operazioni da svolgere.

## **Organizzazione dell'ambiente domotico EUD.**

La teoria delle attività sottolinea il fatto che ci sono delle mediazioni tra soggetti e oggetti: i soggetti utilizzano l'intelletto o strumenti fisici per le loro attività. Nel contesto EUD per smart homes bisognerebbe raggiungere la giusta mediazione tra gli abitanti e i loro obiettivi, con l'ausilio di strumenti per il controllo diretto dei dispositivi e servizi programmabili dall'utente, quindi, seguire il metodo delle attività per lo sviluppo di tool o framework EUD per le smart homes implica un'organizzazione tale da creare gruppi di attività che un utente potrebbe svolgere all'interno di un ambiente domotico, automatizzando alcune dinamiche per aumentare il confort degli abitanti. Per esempio, uno scenario possibile sarebbe l'organizzazione e l'automatizzazione dell'impianto di illuminazione in una stanza destinata al lavoro, ovviamente questi settaggi dovranno essere differenti se in quella stanza si svolgeranno altre attività. Quindi, i programmi e i sistemi EUD dovrebbero prevedere dei setting dinamici e automatici che riescano a leggere le situazioni dell'ambiente.

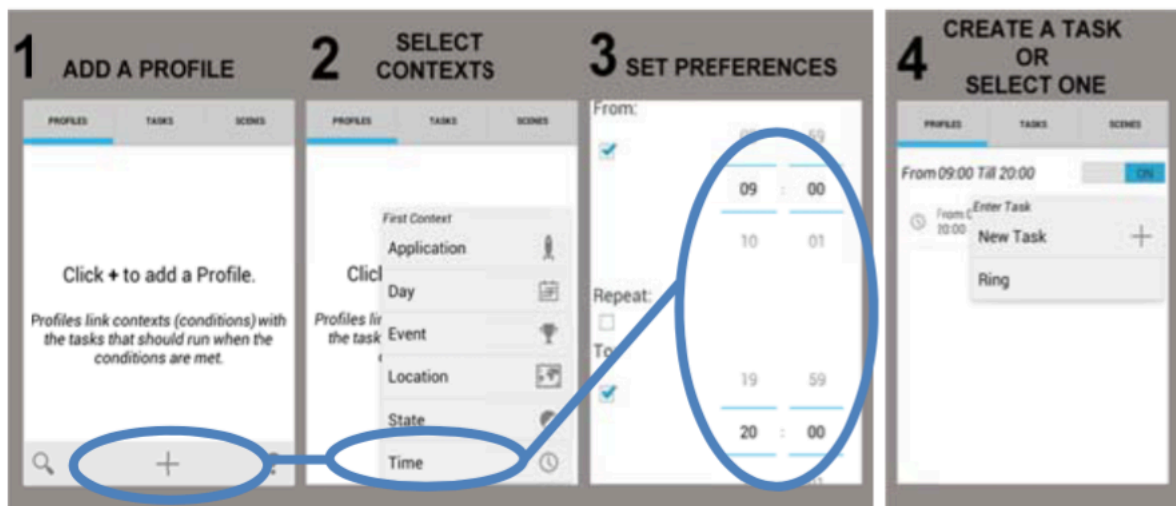
Questo potrebbe significare anche che, a seconda dell'attività, l'interfaccia utente deve subire delle variazioni; per esempio, la UI dovrebbe fornire un controllo diretto dei media se si sta guardando la tv con la famiglia, prevedendo anche meccanismi di condivisione e di migrazione dei servizi su altri dispositivi. Ovviamente, come abbiamo già accennato i tool EUD consentono la creazione dei propri strumenti e UI personalizzate, in modo da creare pannelli di controllo personalizzati così da far

evolvere il tutto in base alle esigenze e alle esperienze degli utenti, la ricerca della Human computer interaction, nel corso degli anni, ha stilato alcune linee guida al fine di aumentare l'interazione e l'esperienza utente.

## **Piattaforme EUD per la domotica (stato dell'arte).**

Possiamo analizzare lo stato dell'arte delle applicazioni EUD, facendo un breve focus su alcune app già accennate nei capitoli precedenti. Le applicazioni in questione sono: Tasker, Locale e Atooma. Successivamente verranno descritte brevemente in modo poi da giungere a delle conclusioni, in cui possiamo fare un incrocio tra quello che la ricerca propone di seguire e ciò che è stato già sviluppato e commercializzato.

### **Tasker.**

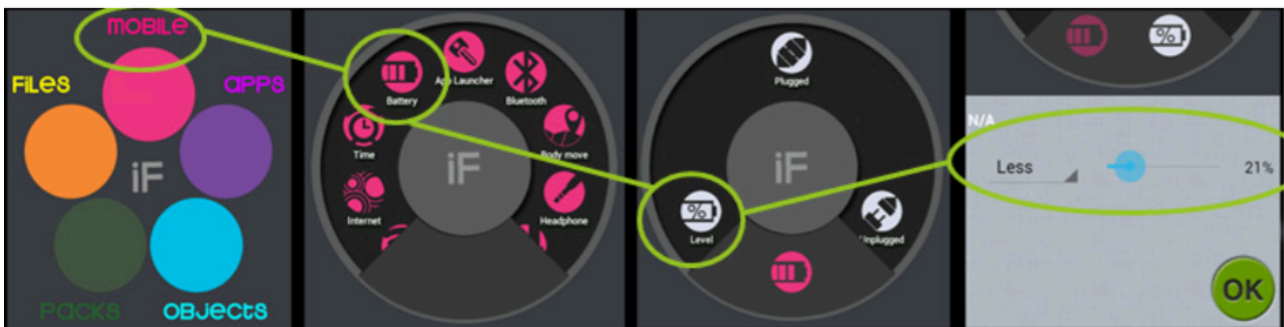


*(interfaccia tasker)*

Tasker è un'applicazione EUD che si basa sul modello azione-condizione-evento; in tasker un'applicazione è chiamata profilo. Per sviluppare un'applicazione, l'utente può scegliere di iniziare con la definizione di una condizione di attivazione, oppure

può iniziare dalla definizione di azioni. Tasker richiede che quando gli utenti specificano una condizione allora devono indicare subito una o più azioni corrispondenti. Tasker permette, inoltre, agli utenti di specificare cosa dovrebbe accadere quando una condizione non è più soddisfatta. Un'ulteriore caratteristica è l'uso di variabili che consente agli utenti di indicare le azioni per i dati sconosciuti in anticipo; ad esempio, visualizzare una notifica dello stato attuale di una connessione o inviare un SMS quando non avete risposto al telefono.

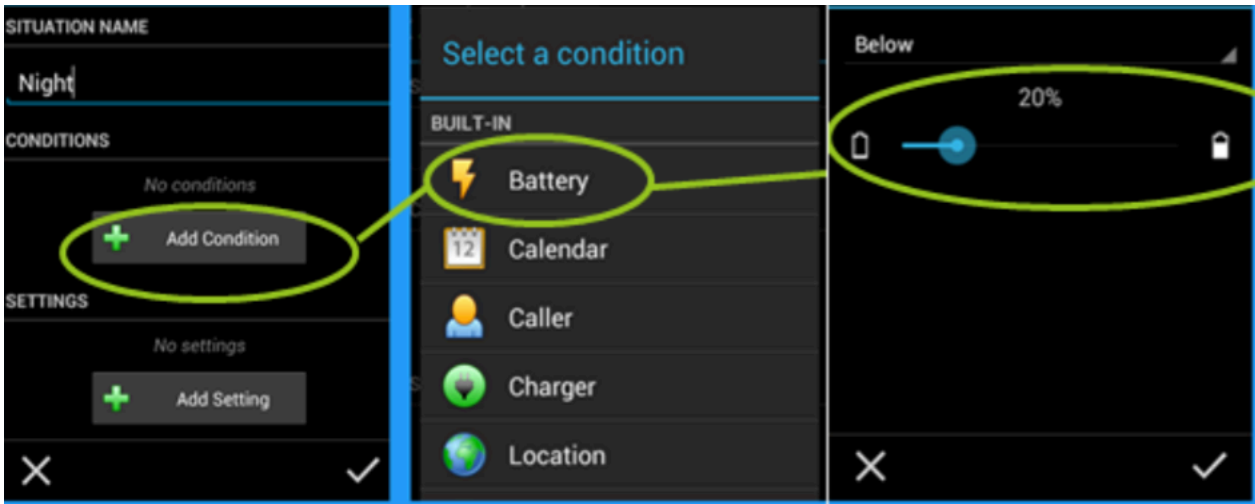
## **Atooma.**



*(ATOOMA UI)*

Atooma rappresenta il modello di condizione / azione con il SE ... DO ... rappresentazione attraverso le rappresentazioni circolari per indicare le categorie e gli elementi rilevanti. La figura mostra come specificare con Atooma se il livello della batteria è inferiore al 20%, esegui una determinata azione. Si richiede quattro passaggi: uno per selezionare la categoria della condizione (mobile), accanto al tipo (batteria), l'attributo (livello) e, infine, il corrispondente valore (meno del 20%). È possibile associare ad atooma applicazioni o dispositivi di terze parti, quindi creare funzioni personalizzate come ad esempio: se arriva una notifica al cellulare cambia la tonalità della luce in cucina.

## Locale.



*(ui locale)*

La caratteristica principale è la sua estensibilità. L'applicazione, infatti, ha un numero limitato di elementi integrati, ma permette agli utenti di estendere le sue funzionalità attraverso l'uso di più di 400 plugin scaricato da Android Market (gratuitamente o pay-per). Il processo di creazione di una situazione non è sequenziale: l'utente può arbitrariamente decidere l'ordine di selezione degli elementi perché l'interfaccia è progettata per visualizzare simultaneamente (cioè sullo stesso schermo) pulsanti per l'aggiunta di condizioni e azioni (in questo ambiente essi sono chiamati Impostazioni), come mostrato in figura.

## Conclusioni.

Le attuali tecnologie sono abbastanza mature per sistemi onnipresenti, ma sistemi onnipresenti nelle case sono latenti. In realtà, il problema è quello di consentire agli utenti di esprimere i loro bisogni e di fornire a loro gli strumenti adatti per soddisfarli,

utilizzando la tecnologia a disposizione nelle loro abitazioni. Lo sviluppo End user development, basato sulle attività, potrebbe integrarsi con il modello condizione-azione-evento con lo scopo di migliorare l'interazione tra abitanti e programmi di domotica. L'obiettivo è quello di arrivare a fornire tutti gli strumenti necessari affinché l'utente possa configurare autonomamente il proprio ambiente domotico al fine di soddisfare le proprie necessità.



## **Bibliografia.**

1. Björkskog, Christoffer. "Human Computer Interaction in Smart Homes."
2. Coutaz, J. et al. (2010) DisQo : A user needs analysis method for smart home in Proceedings of ACM NordiCHI 2010 International Conference, pages 615-618.
3. Davidoff, S. et al. (2006). Principles of smart home control. Lecture Notes in Computer Science, 2006: Ubiquitous Computing (UbiComp 2006), 4206: 19-34.
4. Garcia-Herranz, et al. Towards a Ubiquitous End-User Programming System for Smart Spaces. Journal of Universal Computer Science, 16(12), 012, 1633-1649.
5. Holloway, S., & Julien, C. (2010, November). The case for end-user programming of ubiquitous computing environments. In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 167-172). ACM.
6. Holloway, S., Stovall, D., & Julien, C. (2009). What Users Want from Smart Environments. Technical Report, UT-EDGE-2009-008.
7. Intille, S. (2002). Designing a home of the future. Pervasive Computing, IEEE, pp76-82.
8. Ko, A. J. and Myers, B. A. (2008), "Debugging Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior", pp76-82.
9. Mennicken, S. and Huang, E. M., « Hacking the Natural Habitat : an in-the-wild study of smart homes, their development, and the people who live in them » (2012), Pervasive'12 Proceedings of the 10th international conference on Pervasive Computing, pp143-160.
10. Rogers, Y. 2006. Moving on from weiser's vision of calm computing: engaging ubicomp experiences. In Proceedings of the 8th international conference on Ubiquitous Computing (UbiComp'06), Paul Dourish and Adrian Friday (Eds.). Springer-Verlag, Berlin, Heidelberg, 404-421.
11. Rosen, N., Sattar, R., Linderman, R. W., Simha, R., & Narahari, B. (2004, June). HomeOS: Context-Aware Home Connectivity. In International Conference on Pervasive Computing and Applications.