

POLITECNICO DI TORINO

Facoltà di Ingegneria  
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

**Astrazione automatica di regole  
per sistemi IoT: un approccio  
rivolto ad utenti non esperti**



Relatori:  
Fulvio Corno  
Luigi De Russis

Candidato:  
Fabio Ballati

Marzo 2017

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Dalle regole di basso livello verso un nuovo paradigma</b>	<b>5</b>
2.1	Regole di basso livello, lo stato dell'arte . . . . .	5
2.2	Le ragioni di un paradigma ad alto livello . . . . .	7
<b>3</b>	<b>Analisi del dataset: le regole di basso livello</b>	<b>8</b>
3.1	Il dataset . . . . .	8
3.2	I trigger . . . . .	10
3.3	Le azioni . . . . .	11
3.4	Correlazioni tra trigger ed azioni . . . . .	12
3.5	Commenti relativi al dataset . . . . .	13
<b>4</b>	<b>Alto livello di astrazione: categorie e regole</b>	<b>14</b>
4.1	Categorie e eventi di alto livello . . . . .	14
4.2	Trigger: categorie di alto e medio livello . . . . .	15
4.2.1	Categorie di alto livello . . . . .	15
4.2.2	Categorie intermedie . . . . .	18
4.3	Azioni: categorie di alto e medio livello . . . . .	20
4.3.1	Categorie di alto livello . . . . .	20
4.3.2	Categorie intermedie . . . . .	21
<b>5</b>	<b>Regole azione-reazione: passaggio da basso ad alto livello</b>	<b>24</b>
5.1	Le regole di alto livello . . . . .	24
5.2	Algoritmo per il mapping basso livello - alto livello delle regole . . . . .	27
5.2.1	Obiettivo e logica dell'algoritmo . . . . .	27
5.3	Regole di alto livello: l'algoritmo . . . . .	35
<b>6</b>	<b>Analisi dei dati</b>	<b>61</b>
6.1	Regole con azioni di alto livello . . . . .	61
6.2	Regole con trigger di alto livello . . . . .	62

6.3	Regole di alto livello . . . . .	63
6.3.1	Regole di alto livello considerando autori che hanno pubblicato più di due regole . . . . .	63
6.3.2	Regole di alto livello considerando autori che hanno pubblicato almeno 10 regole . . . . .	64
6.4	Considerazioni . . . . .	67
<b>7</b>	<b>Applicazione web per test utente</b>	<b>69</b>
7.1	Funzionamento . . . . .	69
7.1.1	Utilizzo del paradigma a basso livello . . . . .	70
7.1.2	Utilizzo del paradigma ad alto livello . . . . .	72
7.1.3	Memorizzazione informazioni utente . . . . .	74
7.2	Implementazione . . . . .	75
<b>8</b>	<b>Test utente</b>	<b>76</b>
8.1	Presentazione del test . . . . .	76
8.1.1	Struttura e modalità . . . . .	77
8.1.2	Testo del test . . . . .	78
8.1.3	Regole per svolgere gli scenari nel modo ottimo . . . . .	81
8.2	Presentazione dei risultati . . . . .	83
8.3	Analisi e commento dei risultati . . . . .	89
<b>9</b>	<b>Conclusione</b>	<b>94</b>
	<b>Bibliografia</b>	<b>96</b>

# Capitolo 1

## Introduzione

Internet è uscito ormai da diverso tempo dai confini degli schermi dei computer e anche da quelli di tablet e smartphone, diffondendosi come un liquido in tutto ciò che ci circonda.

L'orologio che utilizziamo riceve le mail e le notifiche dai nostri contatti dello smartphone, l'applicazione "mappe" del cellulare si ricorda al posto nostro dove abbiamo parcheggiato l'auto, negli USA è uscito un dispositivo che tiene il conto di quante uova si hanno ancora nel frigorifero, in modo da poter avere l'informazione "live" mentre si è a fare la spesa.

E' diventata un vero e proprio sistema linfatico del presente, che porta informazioni da noi al mondo e viceversa attraverso gli oggetti che usiamo tutti i giorni. Un traffico impressionante di dati, numeri, parole, posizioni geografiche, immagini, caratteri e tratti della personalità, stati d'animo, relazioni.

L'IoT, o Internet of Things, è il neologismo riferito all'estensione di Internet al mondo degli oggetti e dei luoghi concreti. Attraverso chip e sensori inseriti al loro interno, gli oggetti sono in grado di interagire tra loro e con la realtà circostante. Così il mondo fisico può essere (quasi) interamente digitalizzato, monitorato e in molti casi virtualizzato.

Il numero di dispositivi IoT è in costante crescita. La strada pare ormai segnata e il sorpasso definitivo di tali dispositivi sui cellulari avverrà entro il 2018. E' quanto emerge dall'ultima edizione del Mobility Report di Ericsson[1], secondo cui su un totale di 28 miliardi di dispositivi connessi nel 2022, circa 16 miliardi saranno dispositivi IoT, con un incremento del 23% annuo fra il 2015 e il 2022 [FIG1.1].

L'Europa occidentale farà da apripista nella diffusione dei dispositivi IoT, con un incremento di essi del 400% entro il 2022. Un trend che sarà dovuto principalmente a

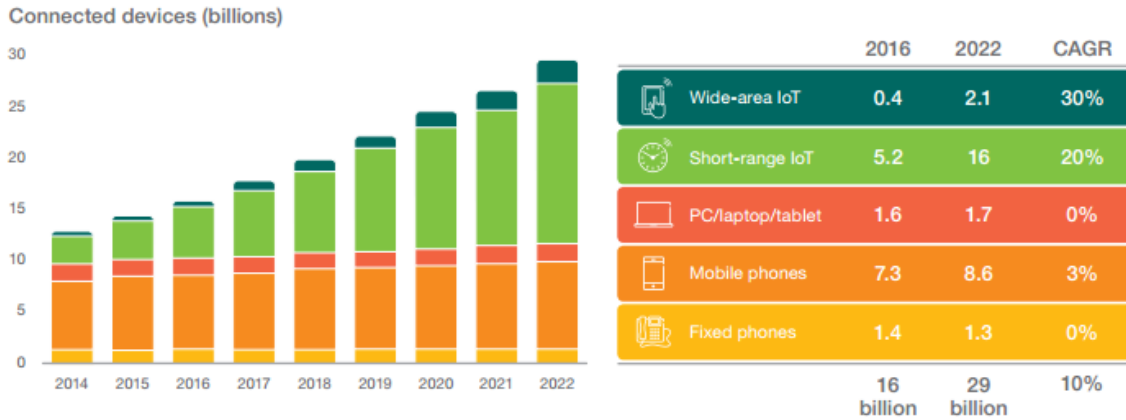


Figura 1.1. Mobility Report: incremento dei dispositivi connessi

questioni regolamentari adottate a livello comunitario, come quelle che obbligheranno ad installare i cosiddetti contatori intelligenti (smart meter) così come il servizio elettronico di chiamata di emergenza (eCall) da installare sulla propria autovettura per limitare i tempi di intervento dopo un incidente.

La costante riduzione dei costi dei dispositivi, nonché la crescente disponibilità di apparecchi da utilizzare in nuovi campi, sta stimolando questo trend e, sempre secondo l'analisi svolta da Ericsson, con l'arrivo delle nuove reti 5G -previsto dal 2020- ci saranno anche le infrastrutture di rete adeguate a stimolare ulteriormente l'adozione dell'IoT.

Questo scambio imponente e costante di informazioni è spesso un groviglio e non pochi percorsi digitali si esauriscono poco dopo l'inizio. E' proprio nella gestione di questi snodi che si inserisce l'idea di servizi (come IFTTT, Microsoft Flow, Tarpipè) pensati come lingua universale per l'IoT così da far interagire tra loro elementi connessi di diversi produttori, anche se ovviamente siamo noi a decidere cosa deve succedere e quando.

Il principio di funzionamento di questi servizi è banale e davvero alla portata di tutti secondo la regola dell'azione / reazione: se succede una determinata cosa (trigger) allora deve scattare una determinata reazione (action). Il tutto tramite semplici regole scritte dagli utenti, da creare partendo da zero oppure da scegliere tra quelle già scritte da altri e personalizzabili, così che tutti questi punti singoli che compongono la nostra vita digitale possano comunicare e interagire, modificando condizioni su internet e nel mondo reale.

Se utilizziamo ad esempio “iOS Location”, ovvero il servizio di localizzazione dell’iPhone, possiamo chiedere a queste piattaforme di spegnere il riscaldamento o le luci Philips Hue quando ci allontaniamo da casa. Le regole che si possono creare sono infinite: c’è chi vuole una notifica se un sito pubblica una notizia, o semplicemente vuole pubblicare in automatico su Facebook la foto appena scattata con l’iPhone.

Il crescente bisogno di personalizzare il funzionamento dei dispositivi IoT da persone che non sono programmatori è quindi attualmente gestito da queste piattaforme basate sulla programmazione per regole. Studi recenti mostrano come questi servizi forzino gli utilizzatori ad usare regole a “basso livello”, ad esempio a livello (utilizzando i termini specifici) del dispositivo / applicazione di cui ci si intende fare uso. Programmare a basso livello comporta dover modellare ogni input come uno specifico evento trigger ed ogni output come un evento azione. Gli utilizzatori preferirebbero invece utilizzare servizi ad un “più alto livello di astrazione”, dove le regole sono più generali, più comprensibili e meno limitanti rispetto a quelle di basso livello le quali assumono la forma di “Esattamente in questo caso, devi fare esattamente questa azione”.

Obiiettivo della tesi è quello di definire, progettare e validare una metodologia che permetta di tradurre automaticamente regole di basso livello, come quelle presenti nei servizi sopra riportati, in una rappresentazione più astratta che l’utente finale possa comprendere e sfruttare con maggiore semplicità e gradimento.

Alcuni delle domande a cui verrà data una risposta sono: quante regole di basso livello esistenti possono essere traslate con successo nelle rispettive regole di alto livello? Quali sono i gruppi, i cluster, le regole di associazione che determinano la similarità delle regole di basso livello? Qual è la percezione degli utenti relativamente le regole astratte rispetto alle corrispondenti regole di basso livello? In caso di disallineamenti nella traduzione, in quali casi possono essere considerate astrazioni errate rispetto alla regola iniziale ed in quali altri casi un miglioramento di questa? Tutte le regole di basso livello possono e dovrebbero essere rese più astratte?

Quelle presentate sono alcune delle domande alle quali si troverà risposta nei capitoli di questo scritto. Si parla di “quante” e “quali” regole in quanto il punto di partenza per lo studio svolto è un dataset contenente regole di basso livello estratte dalle regole pubbliche di IFTTT, presente tra i servizi precedentemente citati nell’ambito della gestione dei dispositivi IoT.

Il dataset e le statistiche da esso estratte saranno descritte nel terzo capitolo di questo elaborato, a seguito della presentazione dettagliata, presente nel prossimo

capitolo, relativa a che cosa si intenda per “regola di basso livello”.

Lo scritto proseguirà con la presentazione del concetto di “alto livello” relativo alle regole ed alle categorie, per poi entrare nel merito dell’algoritmo che esegue il mapping tra le regole dei due paradigmi, alto e basso livello.

Nei capitoli dal 6 all’8 saranno presentati i dati e le valutazioni rispetto all’applicazione del paradigma di alto livello. Tali dati sono ottenuti attraverso due modalità: una serie di dati numerici è presentata come output dell’algoritmo, un’altra parte di dati è presentata come risultato di un test utente svolto per confrontare i paradigmi di alto e basso livello e confermare i dati emersi dall’algoritmo.

L’ultimo capitolo fornisce una visuale complessiva del lavoro e si conclude sottolineando l’intuizione per cui si è proposto il paradigma di alto livello.

## Capitolo 2

# Dalle regole di basso livello verso un nuovo paradigma

In questo capitolo illustrerò lo stato dell'arte relativo alle regole di basso livello e le motivazioni per cui si valuta opportuno considerare un nuovo paradigma di regole ad un livello di astrazione superiore.

### 2.1 Regole di basso livello, lo stato dell'arte

Le regole di basso livello permettono all'utente di automatizzare e personalizzare alcune operazioni che si verificano sempre abbinate a medesime altre azioni. Le regole di basso livello funzionano mediante il paradigma dell'azione / reazione. Questo paradigma è composto da due comportamenti.

Il primo comportamento è solitamente noto come "Trigger", in quanto una volta che questo si è verificato farà scattare la reazione.

Il secondo comportamento, la reazione, è solitamente noto come "Azione", la quale viene eseguita automaticamente una volta che è scattato il trigger.

La regola di basso livello prende la seguente forma:

Ogni volta che si verifica il comportamento TRIGGER, allora esegui il comportamento AZIONE.

Un esempio:

Ogni volta che SONO TAGGATO IN UNA FOTO, allora SALVA LA FOTO  
SUL MIO CLOUD

In dettaglio, notiamo come ogni comportamento debba essere caratterizzato non solo dalla sua propria funzione ma anche da un canale di appartenenza.



La forma completa diventa:

Ogni volta che si verifica il comportamento TRIGGER sullo specifico CANALE di TRIG-

GER, allora esegui il comportamento AZIONE sullo specifico CANALE di AZIONE.

L'esempio precedente risulta essere:

Ogni volta che SONO TAGGATO IN UNA FOTO sul social network FACEBOOK, allora SALVA LA FOTO sul mio DROPBOX.

Gli elementi che compongono questa regola sono:

- Trigger: SONO TAGGATO IN UNA FOTO
- Canale del Trigger: FACEBOOK
- Azione: SALVA LA FOTO
- Canale Azione: DROPBOX.

Attraverso una delle piattaforme per la gestione dei dispositivi IoT l'utente può creare regole di questo tipo selezionando trigger ed azioni a piacere tra quelli delle "cose" supportate dal servizio di gestione scelto. Ciascuna regola per poter essere creata necessita che l'utente dia l'autorizzazione al servizio di accedere ad entrambi i canali relativi ai comportamenti trigger ed azione selezionati. Dopo essere stata creata ciascuna regola necessita che l'utente la attivi. Da quel momento in poi effettivamente la regola funziona in modo autonomo ed automatico, rispondendo con il comportamento azione in seguito al verificarsi del comportamento trigger ad esso associato.

Alcuni esempi di queste regole possono essere:

- Ogni volta che SONO TAGGATO IN UNA FOTO sul social TWITTER, allora CARICA LA FOTO sulla mia CARTELLA GDRIVE.
- Ogni volta che ARRIVO A CASA con la mia auto con all'interno ZUBIE, allora ACCENDI LA LUCE dell'ingresso PHILIPS HUE.
- Ogni volta che SCATTO UNA FOTO con il mio CELLULARE ANDROID, allora PUBBLICALA sul mio profilo di INSTAGRAM.

## 2.2 Le ragioni di un paradigma ad alto livello

Le regole a basso livello vincolano l'utente a ragionare a livello di dispositivo/ servizio e con il linguaggio che ciascuno di questi propone.

Su Facebook l'utente scrive un post, usando Twitter condivide un Tweet. Il "Post" ed il "Tweet" sono entrambi la medesima cosa: del testo pubblicato sul proprio profilo.

A partire da questa constatazione, che in alcuni casi risulta anche essere una limitazione, viene proposta una duplice motivazione che porta a ragionare in termini di regole ad un livello di astrazione superiore.

La prima motivazione è fare in modo che le regole risultino più comprensibili per l'utente. Attraverso il livello di astrazione superiore l'utilizzatore potrà creare regole presentate in un linguaggio simile a quello utilizzato nella vita quotidiana e non sarà vincolato a ragionare a livello dei singoli dispositivi coinvolti nella regola. Le regole così create corrisponderanno ad azioni della vita reale. Queste ovviamente saranno eseguite nel concreto dai medesimi dispositivi e servizi utilizzati per le regole di basso livello, con la differenza che l'utente sceglierà il servizio che più si allinea con l'azione concreta desiderata. Nel paradigma a basso livello si parte dal punto di vista opposto, pensando direttamente in termini di quale applicazione svolga la funzione più simile a quella che è la mia intenzione nella vita concreta.

La seconda motivazione vede un risparmio da parte dell'utilizzatore rispetto al numero di regole create ed al tempo utilizzato per realizzarle. L'obiettivo è quello di aggregare in un'unica regole di maggiore astrazione, più regole di basso livello così da ottenere i sopracitati risparmi da parte dell'utente.

Per comprendere nel concreto l'influenza del livello di astrazione superiore per le regole su questi due aspetti è stato svolto un test utente che sarà descritto nell'ultimo capitolo.

Nei capitoli seguenti saranno esplicitati le analisi dei dati e ragionamenti che hanno condotto all'individuazione delle regole di alto livello. Saranno presentate tali regole e confrontate con quelle di basso livello, prima teoricamente ed infine attraverso il test proposto agli utenti.

# Capitolo 3

## Analisi del dataset: le regole di basso livello

In questo capitolo illustrerò la struttura del dataset utilizzato come base dati per lo svolgimento dell'intero documento. In seguito presenterò i risultati ottenuti interrogando il dataset. Tali dati sono relativi ad un'analisi del dataset considerando le azioni ed i trigger separatamente e per ultimo considerando azioni e trigger assieme per studiare le relazioni che li legano.

### 3.1 Il dataset

Il dataset di riferimento è quello messo a disposizione dal progetto UPOD (“Universal Programming Of Devices”) una ricerca accademica della Brown University [3]. In questo sono presenti 224,590 regole usate dal servizio web IFTTT, il principale tra quelli che si occupano di gestione di dispositivi e applicazioni basata sulla programmazione ad eventi.

Sul servizio IFTTT era possibile (non lo è più da fine 2016) creare regole nelle due forme: pubblica e privata. Nel primo caso le regole create dal singolo utente erano visionabili da ciascun utente.

Il dataset è composto solamente dalle regole pubbliche ed è datato settembre 2015.

Per avere un punto di riferimento comune sul quale poter proporre le argomentazioni riporto di seguito l'esempio di una regola IFTTT presa dal dataset analizzato:

1. id: 59528
2. description: Dropbox my Instagrams (User-provided text that serves as the recipe's title)
3. author: lehmann\_heat (The user who created the recipe)

4. date: September 29, 2012 (Date of posting)
5. shares: 520 (How many other users have added the recipe to “my recipes”)
6. triggerchannel: Instagram (The device or service)
7. trigger: Any new photo by you
8. triggerdescription: This Trigger fires every time you share any new photo on Instagram. (An IFTTT-generated description of the trigger)
9. actionchannel: Dropbox
10. azione: Add file from URL
11. actiondescription: This Action will download a file at a given URL and add it to Dropbox at the path you specify. NOTE: 30 MB file size limit. (An IFTTT-generated description of the action)

La regola del servizio web IFTTT, basato sulla programmazione trigger-action, rappresentata dai dati soprariportati si legge nel modo seguente:

- Se (if) scatta un Trigger (this) di un certo triggerchannel
- Allora (then) esegui l’azione (that) su uno specifico actionchannel

Segue un esempio concreto a partire dai dati presentati:

- Se io pubblico una nuova foto su Instagram
- Allora il file presente alla URL specificata viene scaricato ed aggiunto su Dropbox nella cartella che io ho specificato

I tre campi avente il termine “description” all’interno del nome sono stati ignorati nell’analisi in quanto contenenti una stringa utile all’utente per una maggiore comprensione della regola ma inutile per fini statistici.

Per ciascun canale di azione e canali di trigger ci sono un certo numero di eventi ad esso associati (rispettivamente azioni e trigger). Al contrario, ciascun comportamento (trigger o azione) è identificabile dal proprio nome e dal canale di appartenenza. Dopo aver illustrato il dataset e la sua struttura inizieremo con l’analisi vera e propria dei dati che vi sono all’interno, trattando le azioni, i trigger e per ultimo le relazioni tra essi.

## 3.2 I trigger

La base dati utilizzata presenta tre informazioni riguardanti i trigger. Le due su cui mi soffermo sono i campi “triggerchannel” e “trigger” mentre il campo “trigger description” è solamente una descrizione non utile a fini statistici.

Nel dataset vi sono 177 canali di trigger, ciascuno composto da un numero diverso di eventi possibili per una somma totale ottenuta da tutti i canali di 697 tipi diversi di possibili trigger.

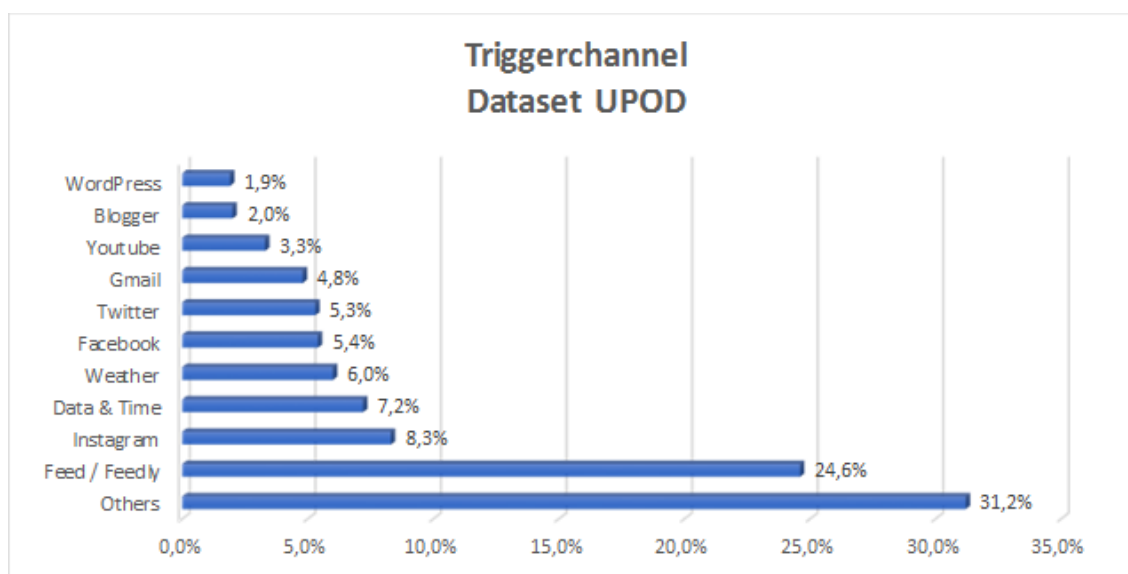


Figura 3.1. Dataset: percentuale di presenza relativa i canali di trigger

Il grafico sopra riportato rappresenta le percentuali in cui i canali di trigger sono presenti nel dataset e quindi sono scelti come evento trigger scatenante un’azione.

Il primo aspetto da sottolineare è come circa il 70% delle regole presenti nel dataset abbiamo come trigger un evento contenuto nei 10 canali di trigger più presenti. In particolare possiamo osservare come l’aggregatore di notizie Feedly sia il trigger più scelto dagli utenti, circa un quarto delle regole hanno esso come evento trigger. Questo dato potrà tornare utile quando si dovrà applicare una suddivisione dei trigger per categorie. Il restante 30% dei dati contenuti nel dataset contiene trigger appartenenti ai rimanenti 167 canali di trigger.

Un altro aspetto importante da evidenziare sono i 67 trigger differenti appartenenti ai 10 canali più presenti (confrontati con il totale di 697 trigger del dataset). Questo dato mostra come ciascuno dei canali di trigger più rappresentativi abbia in media tra i 6 ed i 7 trigger a cui corrisponde un’azione di risposta. In questo modo

traspare come essi siano applicativi più complessi e vari (in termini di interazione con l'utente) rispetto al rimanente 30% dei canali di trigger i quali hanno in media dai 3 ai 4 eventi trigger.

### 3.3 Le azioni

Per quanto riguarda le azioni procederò con un'analisi del tutto simile a quella relativa agli eventi trigger. In questo modo anche solamente leggendo i dati sarà possibile un primo confronto non ancora specifico e dettagliato.

Come per i trigger anche per le azioni sono presenti nel dataset le medesime tre informazioni a riguardo, riportate per completezza. Vi sono i campi "actionchannel" ed "action" di cui a breve parleremo ed il campo "actiondescription" il quale è nuovamente una descrizione dell'azione creata da IFTTT, non utilizzata nell'analisi dei dati.

Nel dataset vi sono 143 canali di azione, ciascuno composto da un numero diverso di azioni scaturenti da trigger, per una somma totale ottenuta da tutti i canali di 303 tipi diversi di possibili azioni.

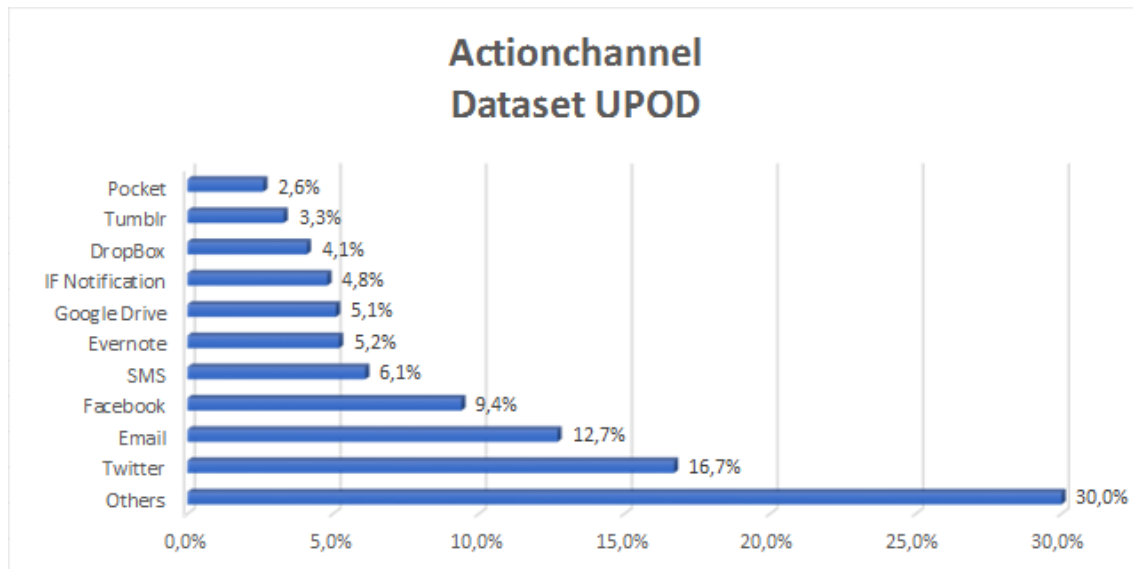


Figura 3.2. Dataset: percentuale di presenza relativa i canali di azione

Il grafico rappresenta le percentuali in cui i canali di azione sono presenti nel dataset e quindi saranno l'azione che si verifica in risposta al trigger ad essa associato.

Per quanto riguarda le azioni sottolineiamo come, anche in questo caso circa il 70% delle regole presenti nel dataset abbiamo l'azione in risposta al trigger contenuta nei

10 canali di azione più presenti.

Guardando alle percentuali di presenza dei canali di azione nel dataset osserviamo come lo stacco tra il canale di azione più presente e gli altri sia minore. Il social network Twitter e il canale di risposta ai trigger più utilizzato (16,7% dei casi), le Email che sono il secondo canale sono distanti solamente 4 punti percentuali, molto meno rispetto alla differenza di 16 punti percentuale tra Feedly ed Instagram per quanto riguarda i trigger.

Proseguendo con l'analisi parallela a quanto fatto per i trigger si nota che i 10 canali di azione più rappresentati nel dataset abbiano un totale di 32 azioni possibili da eseguire in risposta al trigger che le ha azionate. Per quanto riguarda le azioni questo dato mostra come la distribuzione del numero di queste nei canali di azione sia molto simile tra quelli più o meno presenti.

### 3.4 Correlazioni tra trigger ed azioni

Le regole presenti nel dataset sono molto varie, coinvolgono molti dispositivi e servizi differenti.

Sono però presenti alcune regole che ricorrono più volte. Tali regole sono relative ai servizi più utilizzati attualmente: i social, i blog ed i servizi web per la memorizzazione di segnalibri relativi alle letture sul web da parte dell'utente.

Nello specifico è ricorrente trovare utenti i quali hanno creato una serie di regole che permetta di pubblicare su Facebook, Instagram e Twitter tutto ciò che viene pubblicato su uno dei tre servizi singolarmente, con l'obiettivo di avere le stesse informazioni presenti sui tutti e tre i social.

Un altro esempio di regole ricorrenti sono quelle relative la memorizzazione di articoli che compaiono su diversi blog. Un numero non trascurabile di utenti crea regole con l'obiettivo di memorizzare sul proprio servizio cloud (Dropbox, Google Drive, Box) un articolo appartenente ad uno specifico argomento comparso su uno dei tanti blog che segue, relativi a piattaforme per creare blog online differenti (WordPress, Blogger, Weebly).

Sono inoltre presenti regole ricorrenti, anche se in numero minore, per quanto riguarda la gestione dei dispositivi nella casa. Tra queste sono ricorrenti regole che accendono/ spengono alcuni dispositivi (luci/ televisione) quando sta per iniziare una partita di una certa squadra oppure ad una certa ora della giornata.

## 3.5 Commenti relativi al dataset

Il dataset a disposizione è relativo al settembre del 2015.

Nella parte introduttiva ho sottolineato come i dispositivi IoT siano in un periodo di grande crescita. Questo dato è utile per comprendere meglio i dati estrapolati dal dataset.

La maggior parte di trigger ed azioni è attualmente relativa a social, piattaforme di blog, smartphone; probabilmente se si analizzasse un dataset attuale (marzo 2017) i dati sarebbero in parte differenti, le percentuali di trigger ed azioni sarebbero in crescita per i dispositivi IoT in quanto anche il loro utilizzo è in netto aumento.

Dopo aver fatto una pura analisi relativa al contenuto della base dati utilizzata per questo lavoro, il prossimo capitolo focalizza l'attenzione su un paradigma di lettura dei dati presenti ad un livello di astrazione più alto, con l'obiettivo di individuare un ragionamento che permetta di creare raggruppamenti contenenti eventi differenti appartenenti ad un "ambito affine".



# Capitolo 4

## Alto livello di astrazione: categorie e regole

All'interno di questo capitolo ci muoviamo da quello che fino a questo punto è stato chiamato “basso livello” verso un livello maggiore di astrazione per gli eventi che prenderà anche il nome di “alto livello”.

Obiettivo ultimo dello studio è quello di identificare regole di alto livello che racchiudano al loro interno più regole di basso livello.

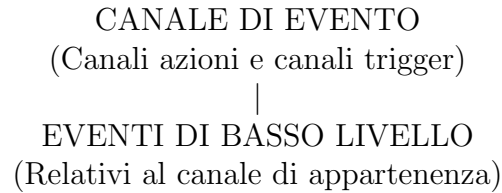
I canali di comportamento presenti nel dataset rappresentano ciascuno un servizio fruibile tramite il web. Nel contesto attuale esistono numerosi servizi che offrono agli utenti medesime o simili possibilità. A partire da tale constatazione, in questo capitolo saranno definite “categorie di alto livello” ed “eventi di alto livello” per i trigger e per le azioni. È possibile vedere questo come il primo passo, fondamentale, per raggiungere l'obiettivo di creare regole di alto livello.

### 4.1 Categorie e eventi di alto livello

La prima necessità è quello di avere un vocabolario comune e comprendere cosa siano categorie ed eventi di alto livello.

Lo schema sottostante aiuta a capire cosa intendiamo per categorie ed eventi di alto livello e come questi si relazionano con gli eventi (trigger ed azioni) di basso livello ovvero quelli direttamente deducibili dai servizi.

RAGIONAMENTO (ATTUALE) DI BASSO LIVELLO:



RAGIONAMENTO PROPOSTO DI ALTO LIVELLO:



A questo punto risulta più semplice comprendere le definizioni seguenti relative le “categoria di alto livello” ed “evento di alto livello”.

Le categorie di alto livello (illustrate ai punti 4.2 e 4.3) hanno la caratteristica di racchiudere eventi, non si ragiona più per canali, appartenenti alla stessa categoria. Gli eventi di alto livello (illustrati nei punti 4.2 e 4.3) faranno riferimento ad una specifica categoria di alto livello.

Ciascuno evento di alto livello ha la caratteristica di racchiudere al suo interno “eventi di basso livello” simili, dove con il termine “simili” si vogliono indicare eventi che svolgono la medesima azione concettuale anche se declinata in termini diversi a seconda del servizio che la fornisce.

## 4.2 Trigger: categorie di alto e medio livello

Sono presentate adesso le categorie e gli eventi di alto livello individuati per i trigger, a seguire sarà introdotto il concetto di categoria intermedia di cui non si è ancora parlato.

### 4.2.1 Categorie di alto livello

Relativamente le categorie dei trigger è presente una breve descrizione con l'ambito di presentare l'ambito di appartenenza.

I trigger presenti nel dataset sono raggruppati in 13 categorie appartenenti a 4 aree nettamente diversificate:

1. I trigger relativi i dati e le informazioni dell'utente.
2. I trigger relativi le relazioni con altre persone sulla rete da parte dell'utente.
3. I trigger relativi a dispositivi che appartengono all'utente.
4. I trigger relativi il mondo in cui l'utente vive e si muove.

All'interno della prima area sono presenti 6 macro-categorie ciascuna di esse rappresenta un diverso ambito in relazione alle azioni dell'utente:

1. Cloud: spazio di archiviazione dei documenti dell'utente online.
2. Calendar / Task: gestione delle informazioni riguardanti eventi programmati in calendario ed impegni o azioni da svolgere da parte dell'utente
3. Personal: visualizzazione di caratteristiche proprie dell'utente e pianificazione di obiettivi personali riguardo la propria persona e il proprio modo di vivere.
4. Media: gestione dei contenuti multimediali online e delle preferenze su questi, propri dell'utente.
5. Entertainment: spazio dedicato ad interessi, hobby e curiosità per il tempo libero.
6. My Business: luogo in cui l'utente si occupa dei propri acquisti, vendite e gestione dei propri beni economici.

Nell'area riguardante le relazioni con altri utenti della rete e presente la categoria Social che racchiude le informazioni relative le iterazioni tra utenti della rete e tra i contenuti che essi condividono.

Nella terza area individuata sono presenti tre categorie relativi a tre tipologie di dispositivi posseduti dall'utente:

1. My device: gestione di impostazioni, iterazioni, informazioni scambiate ed ottenute attraverso uno dei dispositivi mobile appartenente ad un utente specifico.
2. Smart Home: spazio in cui agiscono i dispositivi caratteristici di una casa Smart e connessi alla rete per scambiarsi/ scambiare informazioni con altri dispositivi.

3. Car: visualizzazione delle informazioni provenienti dall'automobile e dagli spostamenti che l'utente attua con essa.

Nell'ultima area individuata sono presenti 3 categorie relative a tre informazioni ricavabili dall'ambiente in cui l'utente si trova in un dato momento:

1. Location: informazione circa la posizione attuale dell'utente e riguardo i suoi spostamenti sul territorio.
2. Date & Time: informazione a riguardo un certo orario ed una certa data o relativi ad intervalli di tempo regolari e ripetuti.
3. Weather: memorizzazione e previsione del tempo e delle condizioni metereologiche nei luoghi di interesse dell'utente.

La tabella sottostante riporta un riassunto grafico delle categorie individuate ed include per ciascuna di esse gli eventi di alto livello di appartenenza.

Cloud	Calendar / Task	Social	My Devices
New file in my storing	Task completed	Photo is uploaded	Contact is added
Photo is uploaded/ stored	Event is started	Photo tagged	Phone call
New reading item is added in my folder	Task is added	Social post is uploaded	Wi-Fi
New reading item is added by me	Workgroup goal/ task is added	Post is uploaded	Battery
New liked article	Event is added	Like / voted	Power Disconnected
New tagged article	Workgroup task completed	New status	New Photo
New item archived	Workgroup task ended	New mention of you	Set mode
New comment by someone	Workgroup task started	New follower	New Screenshot
	Task is removed	Video is uploaded	Bluetooth
		New set	
Smart Home	Entertainment	Your profile changes	Personal
Security	Space	Private message is received	Vital signs
Thermostat	Sport	Private message is sent	Movement
Sensor	The New York Times	Message in a group/ chat is received	Monitored value above
Air features	Stocks		Monitored value below
Cost and saving	Cooking	Car	Physical activity goal is met
Light	Asking	Car is turned on	Habit activity goal is met
Button	Pet	Car is turned off	Weight
Gardening	Wikipedia	Low level fuel	Sleep
Switch Device	App-store	Low level battery	Activity summary
Cooking	Surfing	Car alert	Physical goal not achieved
Door	IFTTT	Driven alert	Habit goal not achieved
Power	Fantasy Sport	Car light	
Noise	Giphy	Event is started	My Business
Brightness		Event is finished	Item in favourite / shopping list
Connect Home Device Generic			New item in shop
Date & Time	Location	Media	Product availability changes online
Every	Enter from an area	Track added to playlist	Shipment status changes
When	Exit from an area	New track	Product price changes
	Enter / exit from an area	New played track	New payment
Weather		New playlist / mix / album	New refund
Weather changes		New artist	
Report		New video	

Figura 4.1. Trigger: categorie ed eventi di alto livello

## 4.2.2 Categorie intermedie

Si introduce il concetto di “categoria intermedia”. Questa si pone come via di mezzo tra la categoria di alto livello ed il trigger di alto livello.

Esempi di categorie intermedie sono quelle su sfondo colorato nella categoria “Smart Home”.

Una categoria di alto livello può contenere trigger di alto livello oppure categorie intermedie. Ciascuna categoria intermedia può contenere solo trigger di alto livello.

Queste immagini chiariscono il concetto

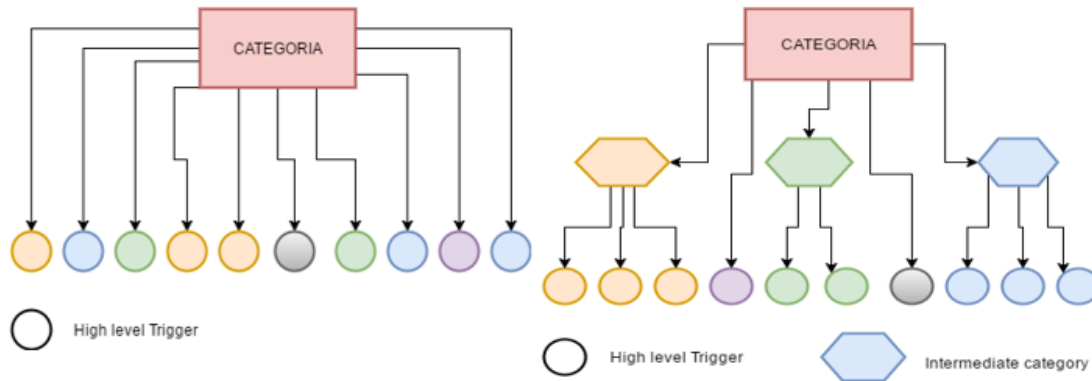


Figura 4.2. Trigger: le categorie intermedie

Il pensiero alla base di quanto mostra l’immagine è quello di passare dalla rappresentazione precedentemente presentata nella quale le categorie di alto livello contengono al loro interno regole di alto livello (Fig sx); ad una suddivisione in cui all’interno delle categorie siano presenti sia regole di alto livello che categorie-intermedie, le quali contengono al loro interno regole di alto livello (Fig dx).

Si utilizza questo ragionamento in quanto alcune categorie hanno un numero di trigger molto numeroso. Lo scopo di questo ulteriore livello di raggruppamento è quello di rendere più navigabile la ricerca del trigger desiderato da parte dell’utente.

In particolare sono stati inseriti nella macro-categoria “Smart-Home” alcune categorie intermedie che a loro volta racchiudono dentro di sé trigger di alto livello. In questo modo all’interno della categoria “Smart Home” non compare una lista lunghissima e confusa di trigger di alto livello, ma compare un elenco più breve di categorie intermedie, all’interno di ciascuna vi sono i trigger di alto livello relativi a quella sola categoria intermedia, questo porta una notevole semplicità di navigazione.

Riassumendo le categorie intermedie hanno lo scopo di:

- limitare il numero di trigger molto diversi tra di loro presenti nella singola macro-categoria, i quali la rendono difficilmente navigabile
- unire trigger di alto livello anche diversi ma provenienti da dispositivi simili, è comodo per l'utente poter avere una visuale d'insieme per lo specifico servizio.

Nella macro-categoria presa in considerazione, l'utilizzo di categorie intermedie è stato necessario per presentare con più chiarezza i trigger di alto livello in essa presenti. È possibile estendere questo ragionamento anche ad altre categorie in modo da facilitarne la visualizzazione da parte dell'utente.

Un esempio è rappresentato dalle categorie intermedie della categoria “Smart-Home”, mostrato nella figura sottostante

<b>Security</b>	<b>Sensor</b>	<b>Thermostat</b>	<b>Air Features</b>
Arm security system	Motion	Temperature drops below	Humidity drops below
Disarm security system	Motion no longer triggered	Temperature rise above	Humidity rises above
Disarm security device	<b>Switch Device</b>	Manage Pause	Air features drops
Arm security device	Switched off	Temperature change	Air features rises above
Alarm triggered	Switched on	<b>Power</b>	
Security device turned change	Locked	Power rises above	
Partially arm security system	Unlocked	Power drops below	
<b>Cost &amp; Saving</b>	<b>Light</b>	<b>Button</b>	<b>Gardening</b>
Cost rises above	Light is switched off	Press	Irrigation event started
Money	Light is switched on		Irrigation event stop
	Long press		Level alert

Figura 4.3. Smart Home: le categorie di trigger intermedie

Come si vede dall'immagine, ciascuna categoria intermedia ha al suo interno un certo numero di trigger di alto livello. Visualizzarli in questo modo e non sotto forma di elenco ne semplifica la comprensione.

Lo stesso ragionamento varrà per le categorie relative le azioni di alto livello; anche in quel caso saranno inserite categorie intermedie nella categoria “Smart-Home”, presa come esempio. Non sarà ripetuto invece il ragionamento relativo alla motivazione di questo nuovo livello.

In conclusione si sottolinea come utilizzando il paradigma ad alto livello il numero di trigger sia stato ridotto, passando dai 697 presenti nel dataset a 140 con la rappresentazione proposta.

Si noti inoltre che non è stato possibile inserire in questi trigger di alto livello 32 dei 697 trigger totali presenti nel dataset. Questo principalmente per due motivi. Alcuni trigger sono relativi a servizi unici per funzionalità. Altri trigger sono marcatamente specifici per un servizio il quale al settembre 2015 (data del dataset) non aveva simili su IFTTT ma erano presenti servizi simili in rete. La situazione attuale invece è diversa, nel contesto attuale su IFTTT ci sarebbero servizi simili e sarebbe possibile incasellare tali eventi in trigger di alto livello.

## 4.3 Azioni: categorie di alto e medio livello

Presento adesso le categorie di alto livello individuati per le azioni, a seguire sarà presentato un esempio di categoria di medio livello relativo alla categoria di alto livello “Smart Home”, come fatto per i trigger.

### 4.3.1 Categorie di alto livello

Per quanto riguarda le categorie è presente una breve descrizione con lo scopo di presentare l’ambito di appartenenza. Le azioni presenti nel dataset sono raggruppate in 7 categorie appartenenti a 3 aree:

1. Le azioni relative i dati e le mie informazioni dell’utente.
2. Le azioni relative le relazioni con altre persone sulla rete.
3. Le azioni relative a dispositivi che appartengono all’utente.

All’interno della prima area sono presenti 4 categorie ciascuna di esse rappresenta una diversa tipologia di informazioni relative all’utente.

1. Cloud: spazio di archiviazione dei documenti dell’utente online.
2. Calendar / Task: gestione delle informazioni riguardanti eventi programmati in calendario ed impegni o azioni da svolgere da parte dell’utente
3. Personal: visualizzazione di caratteristiche proprie dell’utente e pianificazione di obiettivi personali riguardo la propria persona e il proprio modo di vivere.
4. Media: gestione dei contenuti multimediali online e delle preferenze su questi, propri dell’utente.

Nell’area riguardante le relazioni con altri utenti della rete e presente la categoria Social che racchiude le informazioni relative le iterazioni tra utenti della rete e tra i contenuti che essi condividono.

Nell'ultima area individuata sono presenti due categorie relativi a due tipologie di dispositivi posseduti dall'utente:

1. My device: gestione di impostazioni, iterazioni, informazioni scambiate ed ottenute attraverso uno dei dispositivi mobile appartenente ad un utente specifico.
2. Smart Home: spazio in cui agiscono i dispositivi caratteristici di una casa Smart e connessi alla rete per scambiarsi/ scambiare informazioni con altri dispositivi.

La tabella sottostante riporta un riassunto grafico delle categorie individuate ed include per ciascuna di esse gli eventi di alto livello di appartenenza.

<b>Cloud</b>	<b>Calendar/ Task</b>	<b>Smart Home</b>
Add a bookmark to an article / web page	Add a remainder/ todo	Security
Upload photo	Add a task	Light
Upload file	Add event to calendar	Thermostat
Create a multimedia document		Switch Device
Create a document/ text page	<b>Social</b>	Gardening
Create a new note	Create a text post blog	Environment
Add new source	Create a multimedia post blog	<b>My devices</b>
Add row to existing file	Create a text social post	Audio on
<b>Media</b>	Create a multimedia social post	Audio off
Add to play list	Post a message in a group/ chat	Turn on WiFi
Play music	Post a multimedia in group/ chat	Turn off WiFi
Save track	Modify bio profile data	Turn on Bluetooth
Upload track	Upload profile picture	Turn off Bluetooth
<b>Personal</b>	Add user to list	Print PDF
Log a meal	Create a status	Print photo
Log a mood	Send me private	Print web article
Log your weight	Send something to someone	Print mail or message

Figura 4.4. Azioni: categorie ed eventi di alto livello

### 4.3.2 Categorie intermedie

Il concetto di “categoria intermedia” è stato introdotto nel paragrafo precedente relativamente ai trigger. Tale concetto è identico per quanto concerne le azioni.

Esempi di categorie intermedie sono quelle su sfondo colorato nella categoria “Smart Home”.

In tale categoria l'utilizzo di categorie intermedie è utile per presentare con più chiarezza i trigger di alto livello. È possibile estendere questo ragionamento anche ad



altre macro-categorie in modo da facilitarne la visualizzazione da parte dell'utente.

Un esempio è rappresentato dalle categorie intermedie della categoria “Smart-Home”, mostrato nella figura sottostante

<b>Security</b>	<b>Switch Device</b>	<b>Thermostat</b>
Arm security system	Switch/ Turn off your home thing	Set temperature
Disarm security system	Switch/ Turn on your home thing	Start temperature mode
Arm security device	Switch/ Turn on/off your home thing	Stop temperature mode
Disarm security device	Intermittence device	
Change alarm mode	Switch on your home thing momentarily	<b>Light</b>
	Unlock	Light on
<b>Gardening</b>	Lock	Light off
Irrigation on	<b>Enviroment</b>	Dim light
Irrigation off	Play scene	Change light color

Figura 4.5. Smart Home: le categorie di azione intermedie

Come si vede dall'immagine, ciascuna categoria intermedia ha al suo interno un certo numero di azioni di alto livello. Visualizzarli in questo modo e non sotto forma di elenco ne semplifica la comprensione.

In conclusione si sottolinea come utilizzando il paradigma ad alto livello il numero di action sia stato ridotto, passando dai 303 presenti nel dataset a 65 con la rappresentazione proposta.

Si noti inoltre che anche per le azioni, come per i trigger, non è stato possibile inserire tutti i 303 eventi di basso livello all'interno delle azioni di alto livello individuate.

Non è stato possibile inserire 53 delle 303 azioni totali presenti nel dataset. Questo i medesimi due motivi già descritti per i trigger. Alcune azioni sono relativi a servizi unici per funzionalità. Altre azioni sono marcatamente specifiche per un servizio il quale al settembre 2015 (data del dataset) non aveva simili su IFTTT ma erano presenti servizi simili in rete.

La situazione attuale invece è diversa, nel contesto attuale su IFTTT ci sarebbero servizi simili e sarebbe possibile incasellare tali eventi in azioni di alto livello.

Questo capitolo ha avuto lo scopo di definire i concetti di categoria di alto livello, evento di alto livello e le relazioni che ci sono tra questi e gli eventi presenti nella base dati precedentemente descritti.

Nel prossimo capitolo si procederà in modo analogo relativamente al concetto di regola di alto livello per poi spostare l'attenzione sugli algoritmi utilizzati per passare dalle regole di basso a quelle ad alto livello.

# Capitolo 5

## Regole azione-reazione: passaggio da basso ad alto livello

La prima parte di questo capitolo descriverà il concetto di regola di alto livello. Nella seconda parte è illustrato l'algoritmo utilizzato per passare dalle regole di basso ad alto livello.

### 5.1 Le regole di alto livello

Così come le regole di basso livello anche le regole di alto livello permettono all'utente di automatizzare alcune operazioni che si verificano sempre abbinate a medesime altre azioni.

Anche il funzionamento delle regole di alto livello segue il paradigma azione / reazione composto dai due eventi: il primo chiamato trigger ed il secondo chiamato azione.

La caratteristica specifica delle regole di alto livello è data dall'essere astratte e generali, quindi non vincolate in prima istanza ad uno specifico dispositivo o servizio.

Per far comprendere meglio il concetto anche qui come nel caso precedente riporto la forma e un esempio di regola di alto livello, per poi continuare il discorso su ciò che è alla base dell'idea di tali regole più astratte.

La regola di alto livello ha la seguente forma:

Ogni volta che si verifica il TRIGGER appartenente alla determinata CATEGORIA di TRIGGER, allora esegui l'AZIONE appartenente alla determinata CATEGORIA di AZIONE.

Un esempio:

Ogni volta che SONO PRESENTE IN UNA FOTO su uno dei miei servizi SOCIAL, allora SALVA LA FOTO su uno dei miei servizi CLOUD.

Gli elementi che compongono questa regola sono:

- Trigger di alto livello: SONO PRESENTE IN UNA FOTO
- Categoria di Trigger: SOCIAL
- Azione di alto livello: SALVA LA FOTO
- Categoria di Azione: CLOUD.

Si può ottenere così la definizione di regola di alto livello.

Una regola di alto livello è dunque una regola basata sul concetto di azione / reazione che utilizza sia trigger che azioni di alto livello.

Per le regole di alto livello compare il concetto di “categoria” e scompare in un primo momento il concetto di canale che era ciò che legava la regola di basso livello allo specifico dispositivo.

In un secondo momento a tale regola verranno associati per ciascun evento di alto livello uno o più eventi di basso livello e quindi servizi che la attuano, in tale modo attraverso un’unica regola risulteranno connessi un numero maggiore di dispositivi IoT usati dall’utente.

Riprendendo l’esempio:

Ogni volta che SONO PRESENTE IN UNA FOTO su uno dei miei servizi SOCIAL (Facebook, Twitter, Instagram), allora SALVA LA FOTO su uno dei miei servizi CLOUD (Google Drive, Dropbox).

Gli elementi che compongono questa regola sono:

- Trigger di alto livello: SONO PRESENTE IN UNA FOTO
- Categoria di Trigger: SOCIAL
- Canali di Trigger: Facebook, Twitter, Instagram
- Azione di alto livello: SALVA LA FOTO

- Categoria di Azione: CLOUD
- Canali di Azione: Google Drive, Dropbox.

Gli aspetti caratterizzanti le regole di alto livello, ottenuti mediante la comparsa delle categorie, sono due.

Come primo aspetto non essere vincolati ai singoli servizi nella definizione delle regole.

Come secondo aspetto la possibilità di definire la regola utilizzando un linguaggio che sia più vicino a quello del parlare quotidiano degli utenti perché non legato ai termini specifici dei diversi servizi.

Le regole di alto livello hanno quindi lo scopo di partire da ciò che l'utente ha realmente intenzione di fare per poi solo in un secondo momento declinare questa regola negli specifici canali che la attueranno praticamente. Hanno inoltre come obiettivo quello di permettere all'utente di creare una minor quantità di regole includendo in un'unica regola di alto livello le funzionalità svolte da un numero più ampio di regole di basso livello.

Il paragrafo successivo si muoverà nella direzione del ragionamento ad alto livello, considerando regole di alto livello in quanto aventi contemporaneamente trigger ed azioni di alto livello.

## 5.2 Algoritmo per il mapping basso livello - alto livello delle regole

In questo paragrafo viene illustrato nel dettaglio l'algoritmo che permette di analizzare il dataset guardando alle regole di alto livello considerando quindi trigger ed azioni di alto livello contemporaneamente. Sono descritti l'obiettivo, la logica ed il funzionamento dell'algoritmo.

### 5.2.1 Obiettivo e logica dell'algoritmo

In questo paragrafo è presentato l'algoritmo di mapping in modo discorsivo e conciso, con lo scopo di riuscire ad avere una buona intuizione del ragionamento su cui si basa.

#### Obiettivo dell'algoritmo

L'algoritmo che sarà illustrato sfrutta contemporaneamente la suddivisione di alto livello di trigger ed azioni eseguendo:

- L'analisi del dataset al fine di formare gruppi di regole di basso livello simili ed aggregabili, create da uno stesso autore, aventi:
  - come evento scatenante la regola, un trigger di basso livello che ricada all'interno di uno stesso trigger di alto livello tra quelli individuati nella suddivisione
  - come evento di reazione, un'azione di basso livello che ricada all'interno di una stessa azione di alto livello tra quelle individuate nella suddivisione.
- Il mapping di ciascun gruppo di regole di basso livello simili ed aggregabili, in regole di alto livello. Come output l'algoritmo fornisce, per ciascun autore, le regole di basso livello iniziali, le regole di alto livello create, la differenza della numerosità tra alto e basso livello.

#### Logica dell'algoritmo

L'immagine [Fig5.1] rappresenta i punti principali di cui è composto l'algoritmo. In questo paragrafo ciascun punto è illustrato in modo discorsivo. Nel paragrafo successivo ne viene presente una spiegazione precisa, la quale entra nel dettaglio dei vari step presenti all'interno del singolo punto.

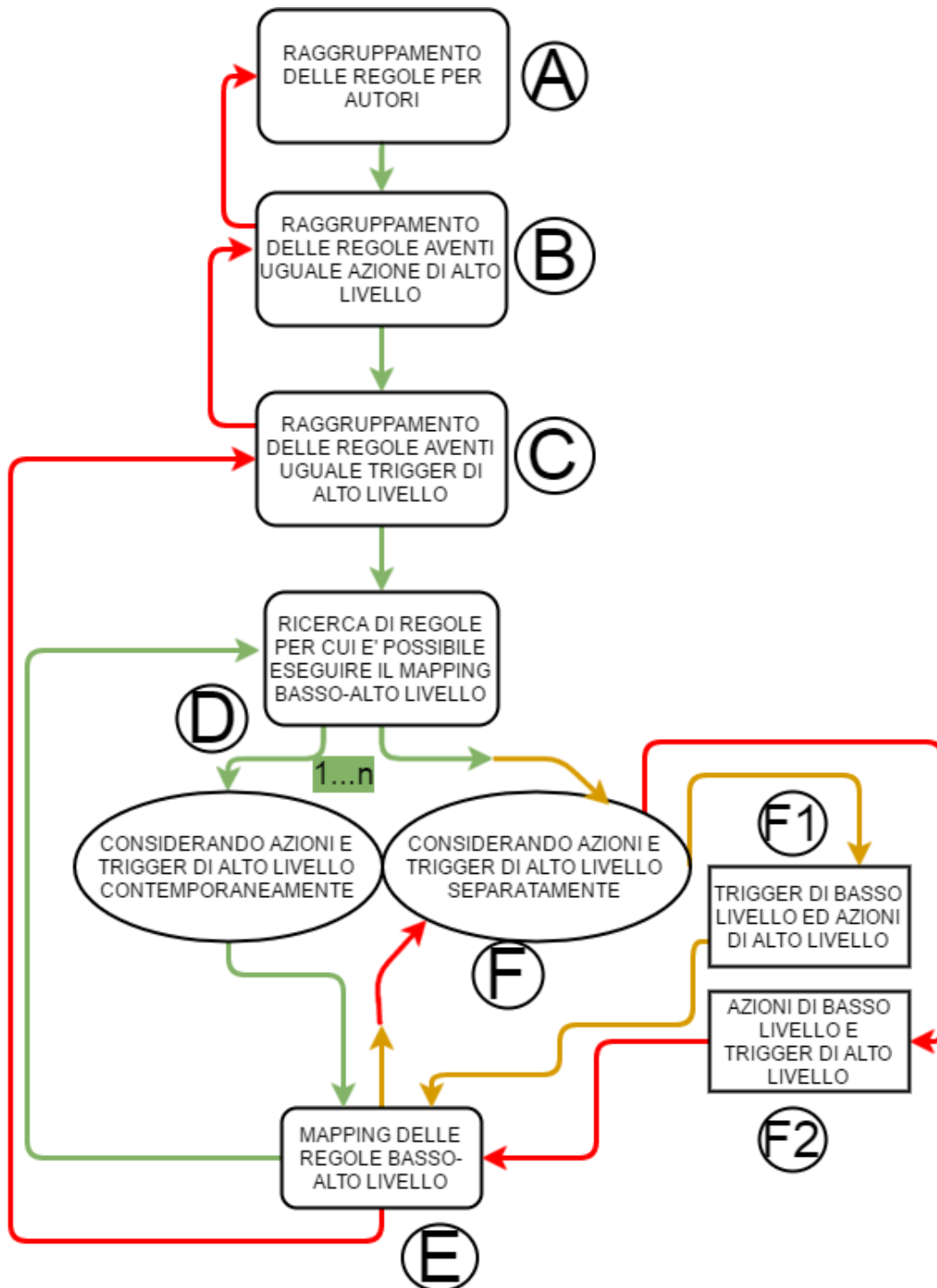


Figura 5.1. Diagramma descrittivo dell'algorithm

**A - RAGIONAMENTO “PER AUTORE”:**

Il punto di partenza su cui si struttura il ragionamento sono i singoli autori delle regole presenti nel dataset.

L’algoritmo suddivide le regole in gruppi, a seconda dell’autore che le ha pubblicate. Input del passaggio B è il gruppo di regole di ciascun autore singolarmente. Il risultato finale potrà quindi essere letto sia per ciascun autore individualmente; sia nel complesso sommando i risultati associati ad ogni autore.

**B - RAGGRUPPAMENTO DELLE REGOLE AVENTI UGUALE AZIONE DI ALTO LIVELLO:**

Per ciascun gruppo di regole appartenenti allo stesso autore, l’algoritmo crea sotto gruppi di regole aventi medesima azione di alto livello. Le possibili azioni di alto livello sono quelle mostrate in precedenza [FIG4.4].

L’algoritmo prende in considerazione un gruppo di regole alla volta come input del punto C. Quando tutti i gruppi di regole formati al punto B sono stati utilizzati come input per C:

- se è già stato eseguito il punto F, l’algoritmo torna ad A considerando quindi un nuovo autore
- altrimenti l’algoritmo esegue il punto F.

**C - RAGGRUPPAMENTO DELLE REGOLE AVENTI UGUALE TRIGGER DI ALTO LIVELLO:**

Per ciascun sotto gruppo individuato al punto B l’algoritmo crea ulteriori sottogruppi di regole aventi medesimo trigger di alto livello. I possibili trigger di alto livello sono quelli mostrati in precedenza [FIG4.4]. Output dei passaggi B e C sono gruppi di regole considerate simili, aventi cioè medesimo trigger di alto livello e medesima azione di alto livello.

Si noti come da questa suddivisione sono escluse le regole aventi trigger ed azioni non appartenenti ad alcun evento (azione o trigger) di alto livello. Esse saranno considerate nel passaggio F.

L’algoritmo prende in considerazione un gruppo di regole alla volta come input del punto D. Quando tutti i gruppi di regole formati al punto C sono stati utilizzati come input per D, l’algoritmo torna a B considerando quindi un nuovo gruppo di regole aventi medesima azione di alto livello.

**D - RAGGRUPPAMENTO DELLE REGOLE PER CUI E’ POSSIBILE ESEGUIRE IL MAPPING:**

All’interno di ciascun gruppo di regole simili l’algoritmo cerca un sottogruppo di regole mappabili. Con il termine “mappabili”, per cui è possibile fare il



mapping basso livello - alto livello, ci si riferisce a gruppi di regole:

- in cui sono presenti almeno due differenti trigger
- per un medesimo evento trigger, sia uguale (anche in termini di numerosità, per ciascuna azione) l'elenco di azioni associate a quel trigger

Quando non è più possibile creare gruppi di regole mappabili, l'algoritmo torna a C considerando quindi un nuovo gruppo di regole simili.

#### E - MAPPING DELLE REGOLE DA BASSO AD ALTO LIVELLO:

A partire da ciascun gruppo individuato in D (oppure in F), l'algoritmo esegue il mapping di più regole di basso livello in una regola di alto livello. Per ciascun gruppo il numero di regole create è equivalente alla numerosità della singola azione.

L'algoritmo aggiorna il numero di dati del gruppo di regole ricevuto come input per poi proseguire al punto D se il passaggio precedente era stato proprio D, altrimenti proseguire con F.

#### F - RICERCA DI GRUPPI DI REGOLE AGGREGABILI CONSIDERANDO SEPARATAMENTE TRIGGER ED AZIONI DI ALTO LIVELLO:

Input di questo passaggio sono le rimanenti regole, quelle aventi trigger ed azioni per cui non è possibile individuare dei gruppi di regole aggregabili che considerino contemporaneamente trigger di alto livello ed azioni di alto livello. In questo passaggio si inseriscono anche le regole aventi trigger o azione non rientranti in alcun evento di alto livello.

Per queste si ragiona in un primo momento in termini di trigger di basso livello, per poi fare il ragionamento duale relativo le azioni.

Se il punto F.1 non è ancora stato eseguito, si esegue F.1, altrimenti F.2

F.1 L'algoritmo forma gruppi di regole aggregabili per cui:

- il trigger è lo stesso per tutte le regole
  - l'elenco delle diverse azioni associate al trigger abbia stessa numerosità
- Si procede con il mapping (E).

F.2 L'algoritmo forma gruppi di regole aggregabili (sono le rimanenti regole di basso livello, non ancora mappate in regole di alto livello) per cui:

- l'azione è la stessa per tutte le regole
- l'elenco dei diversi trigger associate alle azioni abbia stessa numerosità

Si procede con il mapping (E).

E' riportato un esempio che mostra come avviene la creazione delle regole di alto livello senza aggiungere ulteriori spiegazioni rispetto alla descrizione sopra illustrata.

A

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
Facebook	New post by you	Google Drive	Append to a document
Weather	Sunrise	Gmail	Send me an email
Twitter	New tweet by you	Dropbox	Append to a text file
Facebook	New post by you	Google Drive	Append to a document
Android	New message	Google Drive	Append to a document
Weather	Sunset	Gmail	Send me an email
Facebook	New post by you	Dropbox	Append to a text file
Android	New message	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Weather	Sunset	IFTTT	Send a notification
Twitter	New tweet by you	Google Drive	Append to a document
Weather	Sunrise	IFTTT	Send a notification
Facebook	New post by you	Google Drive	Append to a document
Twitter	New tweet by you	Dropbox	Append to a text file
Twitter	New tweet by you	Google Drive	Append to a document

Figura 5.2. Punto A

B

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
GRUPPO1			
Facebook	New post by you	Google Drive	Append to a document
Facebook	New post by you	Google Drive	Append to a document
Facebook	New post by you	Google Drive	Append to a document
Twitter	New tweet by you	Google Drive	Append to a document
Twitter	New tweet by you	Google Drive	Append to a document
Android	New message	Google Drive	Append to a document
Facebook	New post by you	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Twitter	New tweet by you	Dropbox	Append to a text file
Twitter	New tweet by you	Dropbox	Append to a text file
Android	New message	Dropbox	Append to a text file
GRUPPO2			
Weather	Sunset	IFTTT	Send a notification
Weather	Sunset	Gmail	Send me an email
Weather	Sunrise	IFTTT	Send a notification
Weather	Sunrise	Gmail	Send me an email

Figura 5.3. Punto B

C

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
GRUPPO1			
Facebook	New post by you	Google Drive	Append to a document
Facebook	New post by you	Google Drive	Append to a document
Facebook	New post by you	Google Drive	Append to a document
Twitter	New tweet by you	Google Drive	Append to a document
Twitter	New tweet by you	Google Drive	Append to a document
Facebook	New post by you	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Facebook	New post by you	Dropbox	Append to a text file
Twitter	New tweet by you	Dropbox	Append to a text file
Twitter	New tweet by you	Dropbox	Append to a text file
GRUPPO2			
Weather	Sunset	IFTTT	Send a notification
Weather	Sunset	Gmail	Send me an email
Weather	Sunrise	IFTTT	Send a notification
Weather	Sunrise	Gmail	Send me an email
REGOLE ESCLUSE, NON ESISTE TRIGGER DI ALTO PER new message di Android			
Android	New message	Google Drive	Append to a document
Android	New message	Dropbox	Append to a text file

Figura 5.4. Punto C

D

Relativamente il GRUPPO1

	CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
1	Facebook	New post by you	Google Drive	Append to a document
2	Facebook	New post by you	Google Drive	Append to a document
3	Twitter	New tweet by you	Google Drive	Append to a document
4	Twitter	New tweet by you	Google Drive	Append to a document
5	Facebook	New post by you	Dropbox	Append to a text file
6	Facebook	New post by you	Dropbox	Append to a text file
7	Twitter	New tweet by you	Dropbox	Append to a text file
8	Twitter	New tweet by you	Dropbox	Append to a text file

Numerosità della differente singola azione per ciascun trigger = 2

Figura 5.5. Punto D

E

REGOLA DI ALTO LIVELLO EQUIVALENTE ALLE REGOLE 1,3,5,7		
CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
SOCIAL	New social post	Facebook, Twitter
CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
Cloud	Append to a file	Google Drive, Dropbox
REGOLA DI ALTO LIVELLO EQUIVALENTE ALLE REGOLE 2,4,6,8		
CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
SOCIAL	New social post	Facebook, Twitter
CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
Cloud	Append to a file	Google Drive, Dropbox

Figura 5.6. Punto E

F

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
Facebook	New post by you	Google Drive	Append to a document
Facebook	New post by you	Dropbox	Append to a text file
Android	New message	Google Drive	Append to a document
Android	New message	Dropbox	Append to a text file

Figura 5.7. Punto F

F.1

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
GRUPPO 1			
1Facebook	New post by you	Google Drive	Append to a document
2Facebook	New post by you	Dropbox	Append to a text file
GRUPPO2			
3Android	New message	Google Drive	Append to a document
4Android	New message	Dropbox	Append to a text file

Figura 5.8. Punto F.1

E

REGOLA DI ALTO LIVELLO EQUIVALENTE ALLE REGOLE 1,2		
CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
SOCIAL	New social post	Facebook
CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
Cloud	Append to a file	Google Drive, Dropbox
REGOLA DI ALTO LIVELLO EQUIVALENTE ALLE REGOLE 3,4		
CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
SOCIAL	NEW MESSAGE*	Android
CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
Cloud	Append to a file	Google Drive, Dropbox

Figura 5.9. Punto E

Al punto C dell’algoritmo si è detto che le due regole aventi come canale di trigger ”Android”, non avessero associato alcun trigger di alto livello. In questo punto E, il mapping delle regole è invece eseguito. Questo è motivato in quanto è possibile, per ogni evento di basso livello non rientrante in alcuno degli eventi di alto livello, immaginare un evento di alto livello contenente solo l’evento di basso livello escluso dagli altri raggruppamenti. Questo è mostrato in [Fig5.9] dove per gli eventi di basso livello ”new message” sono tradotti con l’omonimo evento di alto livello.

### REGOLE DI ALTO LIVELLO EQUIVALENTI A QUELLE INIZIALI DI BASSO LIVELLO

	CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
1T	SOCIAL	New social post	Facebook, Twitter
2T	SOCIAL	New social post	Facebook, Twitter
3T	SOCIAL	New social post	Facebook
4T	SOCIAL	NEW MESSAGE*	Android
5T	WEATHER	Sun	Weather
	CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
1A	CLOUD	Append to a file	Google Drive, Dropbox
2A	CLOUD	Append to a file	Google Drive, Dropbox
3A	CLOUD	Append to a file	Google Drive, Dropbox
4A	CLOUD	Append to a file	Google Drive, Dropbox
5A	SOCIAL	Send me a notify	IFTTT, Gmail

Figura 5.10. Regole di alto livello equivalenti

## 5.3 Regole di alto livello: l’algoritmo

Il ragionamento dell’algoritmo è ripetuto per ciascun singolo autore ed è composto dai 6 punti visti in precedenza. Di seguito ciascuno dei punti è illustrati nel dettaglio, mostrando eventuali step interni.

### Step 0

Inizializzazione

Inizializzo a zero un contatore “Regole\_risparmiate”.

Inizializzo a zero un contatore “Regole\_altolivello”.

### Punto A

Sono suddivise le regole del dataset in gruppi di regole di basso livello in base all’autore da cui sono state create.

### Punto B e Punto C

#### *Step B&C- 1*

Se sono già stati considerati tutti gli utenti del dataset, si va allo step conclusivo.

Per l’autore attualmente considerato sono suddivise tutte le regole formando gruppi composti da regole aventi il campo azione appartenente ad una medesima azione di alto livello presente tra quelle dei sette ambiti proposti per le azioni ed il campo trigger appartenente ad un medesimo trigger di alto livello presente tra quelli delle tredici categorie di alto livello dei Trigger.

Le regole aventi azione o trigger non appartenente ad alcuna delle categorie di alto livello rispettive sono escluse dai raggruppamenti. Per esse sono create due liste:

- Lista\_trigger\_senza\_categoria\_altolivello, nella quale sono inserite le regole avente trigger non appartenente ad alcuna categoria di alto livello
- Lista\_azioni\_senza\_categoria\_altolivello, nella quale sono inserite le regole avente azione non appartenente ad alcuna categoria di alto livello

Inizializzo a zero un contatore “Regole\_risparmiate\_utente”.

### Punto D

#### *Step D-1*

Se sono già stati considerati tutti i raggruppamenti per l’utente attuale, si prosegue con il punto F.

Si considera un raggruppamento creato nello Step1, si crea una lista ordinata di regole utilizzando come chiave di ordinamento l'unione dei due campi "triggerchannel" + "trigger" (sarà usato fino al termine della descrizione dell'algoritmo il termine trigger intendendo la chiave "triggerchannel" + "trigger" per alleggerire la descrizione).

### **Step D-2**

A partire dalla lista ordinata di regole, per ogni differente trigger creo una coppia <chiave, valore>, chiamata "Trigger-Azioni" avente per chiave il campo trigger e come valore una lista di stringhe, chiamata "Lista\_azioni", il cui ordinamento è dato dal valore proveniente dall'unione dei due campi "actionchannel" + "action" (sarà usato fino al termine della descrizione dell'algoritmo il termine azione intendendo la chiave "actionchannel" + "action" per alleggerire la descrizione).

La struttura dati "Lista\_azioni" ha la seguente struttura:

Lista<azione>

Ciascuna coppia "Trigger-Azioni" ha la seguente struttura:

<trigger, Lista\_azioni>

La struttura dati ottenuta come output dello step3, chiamata "Lista\_Trigger-Azioni", ha la seguente forma:

Lista<Trigger-Azioni>, scritta in forma espansa

Lista< < trigger, Lista<azione> > >

### **Step D-3**

Se sono già stati considerati tutte gli elementi in Lista trigger-azioni, si torna allo step 2.

Per ogni elemento (coppia Trigger-Azioni) della Lista\_Trigger-Azioni, si considera il valore "Lista\_azioni" ed a partire da ciascuna delle diverse azioni presenti in queste strutture dati si crea una nuova lista, chiamata "Lista\_Trigger-Conteggio\_azioni".

La nuova lista è composta da coppie <chiave, valore>, chiamate "Trigger-Conteggio\_azioni".

- La chiave è il campo trigger dell'elemento attualmente considerato.

- Il valore è una lista (chiamata "Lista\_conteggio\_azioni") di coppie <chiave, valore>, chiamate "Conteggio\_azioni", aventi come chiave una delle diverse azioni della "Lista\_azioni" e come valore la numerosità dell'azione considerata all'interno della "Lista\_azioni", chiamata "Numerosità\_azione".

Ciascuna coppia "Conteggio\_azione" ha la seguente struttura:

<azione, Numerosità\_azione>

Ciascuna lista “Lista\_conteggio\_azione” ha la seguente struttura:

Lista<Conteggio\_azione>

Ciascuna coppia “Trigger-Conteggio\_azioni” ha la seguente struttura:

<trigger, Lista\_conteggio\_azione>

La struttura dati ottenuta come output dello step4, chiamata “Lista\_Trigger-Conteggio\_azioni”, ha la seguente forma:

Lista<Trigger-Conteggio\_azioni>, scritta in forma espansa

Lista < < Trigger, Lista < <azione, Numerosità\_azione> > > >

Ovvero una lista di coppie aventi come chiave il trigger e come valore una lista di azioni a cui è associata la numerosità per quel trigger.

#### **Step D-4**

Si ordina Lista\_Trigger-Conteggio\_azioni utilizzando come chiave di ordinamento decrescente la lunghezza della Lista\_conteggio\_azioni. La prima coppia Trigger-Conteggio\_azioni in Lista\_Trigger-Conteggio\_azioni è quindi quella avente lunghezza maggiore di Lista\_conteggio\_azione associata al trigger.

Si inizializza a uno il contatore “Indice\_attuale\_coppia\_considerata”.

#### **Step D-5**

Se sono già stati considerati tutte le coppie presenti nella lista ordinata Lista\_Trigger-Conteggio\_azioni, si torna all’inizio del punto D.

Se non esiste si crea la lista di coppie Trigger-Conteggio\_azioni chiamata Lista\_condizione\_verificata.

Si considera la coppia presente nella lista ordinata Lista\_trigger\_conteggio\_azioni alla posizione indice\_attuale\_coppia\_considerata, questa sarà utilizzata nei punti successivi dell’algoritmo con il nome di “Coppia\_riferimento\_Trigger-Conteggio\_azioni”.

Si scorre Lista\_trigger\_conteggio\_azioni seguendo l’ordinamento e fermandomi alla coppia attualmente considerata (esclusa).

Tra queste, si cercano le coppie Trigger-Conteggio\_azioni per cui la lista delle chiavi di Lista\_conteggio\_azioni sia uguale oppure contenga la lista delle chiavi di Lista\_conteggio\_azioni della Coppia\_riferimento\_Trigger-Conteggio\_azioni.

Si inserisce una copia della coppia Trigger-Conteggio\_azioni che soddisfa la condizione, all’interno della lista Lista\_condizione\_verificata.



Quando si termina di iterare sugli elementi della lista, si aumenta di uno indice\_attuale\_coppia\_considerata. Se Lista condizione verificata è vuota e indice\_attuale\_coppia\_considerata è minore della lunghezza di Lista\_Trigger-Conteggio\_azioni, si torna allo step D-5.

Se Lista condizione verificata è vuota e indice\_attuale\_coppia\_considerata è uguale della lunghezza di Lista\_Trigger-Conteggio\_azioni, si va allo step E-4.

### ***Step D-6***

Si inizializza una lista di coppie Conteggio\_azione, chiamata “Lista\_numerosità\_minima\_azioni”, copia della lista Lista\_conteggio\_azione della coppia Coppia\_riferimento\_Trigger-Conteggio\_azioni.

Si itera su ciascun elemento (Trigger-Conteggio\_azioni) di Lista\_condizione\_verificata. A partire da tale elemento, si itera su ciascun elemento (Conteggio\_azione, in forma esplicita <azione, numerosità azione>) di Lista\_conteggio\_azione.

Si confronta Numerosità\_azione dell’elemento attuale con Numerosità\_azione dell’elemento di Lista\_numerosità\_minima\_azioni avente chiave uguale all’elemento attuale.

Se Numerosità\_azione di elemento attuale è minore di Numerosità\_azione dell’elemento presente in Lista\_numerosità\_minima\_azioni, si sostituisce Numerosità\_azione in Lista\_numerosità\_minima\_azioni dell’elemento considerato con Numerosità\_azione di elemento attuale.

## **Punto E**

### ***Step E-1***

Si calcola “min” che è il valore minimo presente tra i valori della lista Lista\_numerosità\_minima\_azioni. Tale valore rappresenta il numero di regole di alto livello che possono essere create, rappresenta quindi il numero di regole risparmiabili per ciascun azione associata al rispettivo trigger.

Si itera su ciascun elemento (Trigger-Conteggio\_azioni) di Lista\_condizione\_verificata. A partire da tale elemento Trigger-Conteggio\_azioni, si itera su ciascun elemento (conteggio azione, in forma esplicita <azione, numerosità azione>) di Lista\_conteggio\_azione.

Si sottrae a numerosità azione dell’elemento attuale il valore min.

Se numerosità azione dell’elemento attuale è uguale a zero, elimino la coppia conteggio azione attuale.

Si incrementa il contatore Regole\_risparmiate con il valore min.  
Si incrementa il contatore Regole\_risparmiate\_utente con il valore min.

### **Step E-2**

A partire dall'elemento Coppia\_riferimento \_Trigger-Conteggio\_azioni, si itera su ciascun elemento (conteggio azione, in forma esplicita <azione, numerosità azione>) di Lista conteggio azione.

Si sottrae a numerosità azione dell'elemento attuale il valore min.  
Si incrementa il contatore Regole\_risparmiate con il valore min.  
Si incrementa il contatore Regole\_risparmiate\_utente con il valore min.  
Si incrementa il contatore Regole\_altolivello di min, tolgo min ai contatori delle regole risparmiate.

### **Step E-3**

Si sostituisce ciascuna coppia in Lista\_Trigger-Conteggio\_azioni con la coppia avente medesima chiave, presente in Lista\_condizione\_verificata.  
Si sostituisce la coppia in Lista\_Trigger-Conteggio\_azioni con chiave uguale al valore della chiave di Coppia\_riferimento \_Trigger-Conteggio\_azioni.  
Se nella lista aggiornata Lista\_Trigger-Conteggio\_azioni vi sono elementi aventi la lista-valore vuota, elimino tali elementi da Lista\_Trigger-Conteggio\_azioni.  
Si eliminano tutti gli elementi presenti in Lista condizione verificata.  
Se il punto prima di E è stato il punto F, si prosegue, altrimenti si torna allo step D-4.

### **Step E-4**

Lista trigger conteggio azioni non contiene più alcuna coppia avente lista di chiavi relative a Lista conteggio azione uguale o contenuta nella medesima struttura dati estratta da un'altra coppia.  
La condizione attuale è quella di avere una lista di coppie trigger-conteggio azioni rappresentanti regole aventi come trigger il medesimo trigger (chiave della coppia) e come azioni quelle presenti nella lista Lista conteggio azione, valore della coppia.  
Se il punto prima di E è stato il punto F, si torna ad F.  
Se sono già state considerate tutte le coppie trigger-conteggio azioni di Lista trigger conteggio azioni, si torna allo step D-5.

Altrimenti si considera una coppia tra quelli presenti in Lista trigger conteggio azioni. Si elimina da Lista conteggio azione (lista valore della coppia) la/una delle coppie

conteggio azione avente massima numerosità azione. Prima dell'eliminazione si memorizza tale valore in `numerosità_massima_azione`.

### **Step E-5**

Si aggiunge al contatore “`regole_risparmiate`” la sommatoria del campo “`numerosità_azione`” di ciascuno degli  $n$  elementi in Lista conteggio azione.

- `regole_risparmiate` = `regole_risparmiate` + ( “`numerosità_azione_1`” + ... + “`numerosità_azione_n`” )

Si aggiunge al contatore “`regole_risparmiate`” la sommatoria del campo “`numerosità_azione`” di ciascuno degli  $n$  elementi in Lista conteggio azione.

- `regole_risparmiate_utente` = `regole_risparmiate_utente` + ( “`numerosità_azione_1`” + ... + “`numerosità_azione_n`” )

Si incrementa il contatore “`regole_alto_livello`” con il valore contenuto in “`numerosità_massima_azione`”.

Tornare allo step E-4.

## **Punto F**

Se F-1 è già stato eseguito, l'algoritmo esegue F-2.

### **Punto F-1**

#### **Step F-1.1**

La lista `Lista_trigger_senza_categoria_alto_livello` è il caso in cui si ragiona avendo trigger di basso livello ed eventualmente azioni di alto livello.

#### **Step F-1.2**

L'algoritmo crea gruppi di regole aventi medesimo trigger di basso livello ed azione appartenente ad una stessa azione di alto livello.

Se sono stati considerati tutti i gruppi creati, si prosegue con F-2, altrimenti si prosegue con lo step successivo.

#### **Step F-1.3**

Si esegue lo Step E-1. L'algoritmo inizializza i valori necessari in questo passaggio assegnando a “`min`” il valore della numerosità minima dell'azione tra quelle del gruppo considerato attualmente.

Crea la lista `Trigger-Conteggio_azioni`, il trigger e la lista della azioni associate a tale trigger sono quelle del gruppo considerato. Tale elemento è aggiunto a Lista condizione verificata.

### **Punto F-2**

Se F-2 è già stato eseguito, l'algoritmo esegue.

Altrimenti nel contatore “Regole\_risparmiate\_utente” è contenuto il numero di regole che si eviterebbe di creare ragionando con regole di alto livello per l'utente attualmente considerato.

Torno al punto B&C.

#### **Step F-2.1**

La lista Lista\_azioni\_senza\_categoria\_alto\_livello è il caso in cui si ragiona avendo azioni di basso livello ed eventualmente trigger di alto livello.

#### **Step F-2.2**

L'algoritmo crea gruppi di regole aventi medesima azione di basso livello e trigger appartenente ad una stesso stesso trigger di alto livello.

Se sono stati considerati tutti i gruppi creati, si prosegue con F-2, altrimenti si prosegue con lo step successivo.

#### **Step F-2.3**

Si esegue lo Step E-1. L'algoritmo inizializza i valori necessari in questo passaggio assegnando a “min” il valore della numerosità minima del trigger tra quelli del gruppo considerato attualmente.

Crea la lista Trigger-Conteggio\_azioni, l'azione e la lista dei trigger associati a tale azione sono quelle del gruppo considerato. Tale elemento è aggiunto a Lista condizione verificata.

E' riporto un esempio completo del funzionamento dell'algoritmo:

### **Step 0 - inizializzazione**

Regole\_risparmiate = 0;

Regole\_alto\_livello = 0;

### **Punto B e Punto C**

Per l'autore considerato si suddividono le regole formando gruppi di regole aventi il campo azione appartenente ad una medesima azione di alto livello tra quelle delle sette categorie delle azioni ed il campo trigger appartenente ad un medesimo trigger di alto livello presente tra quelli delle tredici categorie dei Trigger.

I rispettivi campi trigger di ciascuna regole del Gruppo 1 [Fig5.11] appartengono al trigger di alto livello “Social post is uploaded” della categoria di alto livello “Social” relativa ai trigger.

I rispettivi campi azione di ciascuna regole del Gruppo 1 appartengono all'azione

Canale Trigger	Trigger	Canale Azione	Azione
Facebook	Post by specific user	IFTTT	Send Notification
Twitter	Tweet by specific user	SMS	Send an SMS
Twitter	Tweet by specific user	IFTTT	Send Notification
App.Net	New post by specific user	IFTTT	Send Notification
Facebook	Post by specific user	Email	Send Email
Facebook	Post by specific user	Twitter	Send direct message
Twitter	Tweet by specific user	Email	Send Email
Twitter	Tweet by specific user	SMS	Send an SMS
App.Net	New post by specific user	Email	Send Email
App.Net	New post by specific user	Email	Send Email
Facebook	Post by specific user	Email	Send Email
Facebook	Post by specific user	Twitter	Send direct message
Facebook	Post by specific user	SMS	Send an SMS
Facebook	Post by specific user	SMS	Send an SMS
App.Net	New post by specific user	IFTTT	Send Notification
Twitter	Tweet by specific user	PushOver	Send High Pry Notify
Twitter	Tweet by specific user	Email	Send Email
Facebook	Post by specific user	SMS	Send an SMS
Facebook	Post by specific user	IFTTT	Send Notification
Facebook	Post by specific user	Email	Send Email
Facebook	Post by specific user	SMS	Send an SMS
Twitter	Tweet by specific user	Email	Send Email
Facebook	Post by specific user	Email	Send Email
Facebook	Post by specific user	Email	Send Email
Facebook	Post by specific user	Email	Send Email
Twitter	Tweet by specific user	IFTTT	Send Notification
App.Net	New post by specific user	SMS	Send an SMS
App.Net	New post by specific user	IFTTT	Send Notification
Facebook	Post by specific user	IFTTT	Send Notification
App.Net	New post by specific user	SMS	Send an SMS
Twitter	Tweet by specific user	SMS	Send an SMS
Facebook	Post by specific user	Twitter	Send direct message

Figura 5.11. Gruppo 1

di alto livello “Send me private notification” della categoria di alto livello “Social” relativa le azioni.

I rispettivi campi trigger di ciascuna regole del Gruppo 2 [Fig5.12] appartengono al trigger di alto livello “Post is uploaded” della categoria di alto livello “Social” relativa ai trigger.

I rispettivi campi azione di ciascuna regole del Gruppo 2 appartengono all’azione di alto livello “Add row to existing file” della categoria di alto livello “Cloud” relativa le azioni.

Canale Trigger	Trigger	Canale Azione	Azione
Blogger	Any new post	Google Drive	Add row to spreadsheet
WordPress	New WordPress post	Google Drive	Append to a document
Blogger	Any new post	DropBox	Append to a text file
Tumblr	Any new post	Evernote	Append to note
WordPress	New WordPress post	Google Drive	Add row to spreadsheet
Blogger	Any new post	Evernote	Append to note
Blogger	Any new post	Box	Append to a text file
WordPress	New WordPress post	DropBox	Append to a text file
Tumblr	Any new post	Google Drive	Append to a document
Blogger	Any new post	Google Drive	Append to a document
Tumblr	Any new post	Google Drive	Add row to spreadsheet

Figura 5.12. Gruppo 2

Si inseriscono nelle lista `Lista_trigger_senza_categoria_alto_livello` eventuali regole aventi trigger non appartenente ad alcuna categoria di alto livello (nell'esempio non sono presenti regole rientranti in questa casistica).

Si inseriscono nelle lista `Lista_azioni_senza_categoria_alto_livello` eventuali regole aventi azione non appartenente ad alcuna categoria di alto livello (nell'esempio non sono presenti regole rientranti in questa casistica).

`Regole_risparmiate_utente = 0;`

### Step D-1

Si considera il raggruppamento 1.

Si crea una lista ordinata secondo il valore “Canale Trigger” + “Trigger” [Fig5.13].

Canale Trigger	Trigger	Canale Azione	Azione
Blogger	Any new post	Google Drive	Add row to spreadsheet
WordPress	New WordPress post	Google Drive	Append to a document
Blogger	Any new post	DropBox	Append to a text file
Tumblr	Any new post	Evernote	Append to note
WordPress	New WordPress post	Google Drive	Add row to spreadsheet
Blogger	Any new post	Evernote	Append to note
Blogger	Any new post	Box	Append to a text file
WordPress	New WordPress post	DropBox	Append to a text file
Tumblr	Any new post	Google Drive	Append to a document
Blogger	Any new post	Google Drive	Append to a document
Tumblr	Any new post	Google Drive	Add row to spreadsheet

Figura 5.13. Gruppo 1 Ordinato per Canale Trigger

### **Step D-2**

Si crea per ciascun diverso trigger una la lista ordinata di azioni “Lista\_azioni” i cui elementi sono le stringhe composte dall’unione dei campi Canale Azione + Azione.

Lista\_azioni1 relativa al trigger “Facebook Post by specific user”:

Email Send Email  
Email Send Email  
Email Send Email  
Email Send Email  
Email Send Email  
Email Send Email  
IFTTT Send Notification  
IFTTT Send Notification  
IFTTT Send Notification  
SMS Send an SMS  
SMS Send an SMS  
SMS Send an SMS  
SMS Send an SMS  
Twitter Send direct message  
Twitter Send direct message  
Twitter Send direct message

Lista\_azioni2 relativa al trigger “Twitter Tweet by specific user”:

Email Send Email  
Email Send Email  
Email Send Email  
IFTTT Send Notification  
IFTTT Send Notification  
PushOver Send High Pry Notify  
SMS Send an SMS  
SMS Send an SMS  
SMS Send an SMS

Lista\_azioni3 relativa al trigger “App.Net New post by specific user”:

Email Send Email  
Email Send Email  
IFTTT Send Notification  
IFTTT Send Notification  
IFTTT Send Notification

SMS Send an SMS  
SMS Send an SMS

Si crea una coppia “Trigger-Azioni” <Canale Trigger +Trigger , Lista\_azioni> per ciascun dei tre trigger.

Trigger-Azioni1 < Facebook Post by specific user, Lista\_azioni1 >  
Trigger-Azioni2 < Twitter Tweet by specific user, Lista\_azioni2 >  
Trigger-Azioni3 < App.Net New post by specific user, Lista\_azioni3 >

Si crea la lista Lista\_Trigger-Azioni composta dai tre elementi Trigger-Azioni appena creati:

Trigger-Azioni1  
Trigger-Azioni2  
Trigger-Azioni3

### ***Step D-3***

Si considera il primo elemento in Lista\_Trigger-Azioni, Trigger\_azioni1.

Si creano le coppie Conteggio\_azioni:

Conteggio\_azioni1: < Email Send Email , 6 >  
Conteggio\_azioni2: < IFTTT Send Notification , 3 >  
Conteggio\_azioni3: < SMS Send an SMS , 4 >  
Conteggio\_azioni4: < Twitter Send a direct message , 3 >

Si crea la lista Lista\_conteggio\_azioni1 composta dai 4 elementi appena creati:

Conteggio\_azioni1  
Conteggio\_azioni2  
Conteggio\_azioni3  
Conteggio\_azioni4

Si crea la coppia Trigger-Conteggio\_azioni < Trigger , Lista\_conteggio\_azioni >  
Trigger-Conteggio\_azioni1: <Facebook Post by specific user , Lista\_conteggio\_azioni1>

Si considera il secondo elemento in Lista\_Trigger-Azioni, Trigger\_azioni2.

Si creano le coppie Conteggio\_azioni:

Conteggio\_azioni1: < Email Send Email , 3 >  
Conteggio\_azioni2: < IFTTT Send Notification , 2 >



Conteggio\_azioni3: < PushOver Send High Pry Notify , 1 >

Conteggio\_azioni4: < SMS Send an SMS , 3 >

Si crea la lista Lista\_conteggio\_azioni1 composta dai 4 elementi appena creati:

Conteggio\_azioni1

Conteggio\_azioni2

Conteggio\_azioni3

Conteggio\_azioni4

Si crea la coppia Trigger-Conteggio\_azioni < Trigger , Lista\_conteggio\_azioni >

Trigger-Conteggio\_azioni2:

<Twitter Tweet by specific user , Lista\_conteggio\_azioni1>

Si considera il terzo elemento in Lista\_Trigger-Azioni, Trigger\_azioni3.

Si creano le coppie Conteggio\_azioni:

Conteggio\_azioni1: < Email Send Email , 2 >

Conteggio\_azioni2: < IFTTT Send Notification , 3 >

Conteggio\_azioni3: < SMS Send an SMS , 2 >

Si crea la lista Lista\_conteggio\_azioni1 composta dai 3 elementi appena creati:

Conteggio\_azioni1

Conteggio\_azioni2

Conteggio\_azioni3

Si crea la coppia Trigger-Conteggio\_azioni < Trigger , Lista\_conteggio\_azioni >

Trigger-Conteggio\_azioni3:

<App.Net New post by specific user , Lista\_conteggio\_azioni1>

Si crea la lista Lista\_Trigger-Conteggio\_azioni, composta dai tre elementi Trigger-

Conteggio\_azioni appena creati:

Trigger-Conteggio\_azioni1

Trigger-Conteggio\_azioni2

Trigger-Conteggio\_azioni3

#### ***Step D-4***

Lista\_Trigger-Conteggio\_azioni ordinata utilizzando come chiave di ordinamento decrescente la lunghezza della Lista\_conteggio\_azioni. Nel caso considerato è già ordinata in quanto Conteggio\_azioni1, valore della coppia Trigger-Conteggio\_azioni1 ha Lista\_conteggio\_azioni1 di lunghezza 4, così come la seconda coppia nella lista,

Trigger-Conteggio\_azioni2. Trigger-Conteggio\_azioni3 ha lunghezza di Lista\_ conteggio\_azioni1 pari a 3, è già corretto che sia ultimo.

Indice\_attuale\_coppia\_considerata = 1 ;

### **Step D-5**

Si considera la coppia presente in Lista\_Trigger-Conteggio\_azioni all'indice Indice\_attuale\_coppia\_considerata, essa diventa Coppia\_riferimento\_Trigger-Conteggio\_azioni.

Coppia\_riferimento\_Trigger-Conteggio\_azioni = Trigger-Conteggio\_azioni2;

Si scorrono gli elementi di Lista\_Trigger-Conteggio\_azioni dall'inizio ad Coppia\_riferimento\_Trigger-Conteggio\_azioni:

Primo elemento: Trigger-Conteggio\_azioni1

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento\_Trigger-Conteggio\_azioni:

Email Send Email

IFTTT Send Notification

PushOver Send High Priority Notify

SMS Send an SMS

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-Conteggio\_azioni1 :

Email Send Email

IFTTT Send Notification

SMS Send an SMS

Twitter Send a direct message

La prima lista non è uguale oppure contenuta nella seconda.

Non si hanno altri elementi su cui iterare poiché il successivo elemento in Lista\_Trigger-Conteggio\_azioni è quello preso come riferimento.

Si incrementa Indice\_attuale\_coppia\_considerata: Indice\_attuale\_coppia\_considerata ++;

La lista Lista\_condizione\_verificata è vuota e indice\_attuale\_coppia\_considerata è minore della lunghezza di Lista\_Trigger-Conteggio\_azioni, si torna allo Step D-5.

### **Step D-5**

Coppia\_riferimento\_Trigger-Conteggio\_azioni = Trigger-Conteggio\_azioni3;

Primo elemento: Trigger-Conteggio\_azioni1

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento  
\_Trigger-Conteggio\_azioni:

Email Send Email

IFTTT Send Notification

SMS Send an SMS

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-  
Conteggio\_azioni1 :

Email Send Email

IFTTT Send Notification

SMS Send an SMS

Twitter Send a direct message

La prima lista è contenuta nella seconda.

```
Lista_condizione_verificata.Add(Trigger-Conteggio_azioni1);
```

```
Indice_attuale_coppia_considerata ++;
```

Secondo elemento: Trigger-Conteggio\_azioni2

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento  
\_Trigger-Conteggio\_azioni:

Email Send Email

IFTTT Send Notification

SMS Send an SMS

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-  
Conteggio\_azioni2:

Email Send Email

IFTTT Send Notification

PushOver Send High Pry Notify

SMS Send an SMS

La prima lista è contenuta nella seconda.

```
Lista_condizione_verificata.Add(Trigger-Conteggio_azioni2);
```

```
Indice_attuale_coppia_considerata ++;
```

La lista Lista\_condizione\_verificata non è vuota proseguo con lo Step D-6.

### **Step D-6**

Lista < Conteggio\_azione> Lista\_numerosità\_minima\_azioni:  
Email Send Email , 2  
IFTTT Send Notification , 3  
SMS Send an SMS , 2

Si itera su ciascun elemento di Lista\_condizione\_verificata:

Primo elemento: Trigger-Conteggio\_azioni1

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:

Primo elemento: Conteggio\_azioni1 : < Email Send Email , 6 >

Email Send Email (Conteggio\_azioni1) > Email Send Email (Lista\_numerosità\_minima\_azioni(0))

6 > 2 quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 3 >

IFTTT Send Notification (Conteggio\_azioni2) = IFTTT Send Notification (Lista\_numerosità\_minima\_azioni(1))

3 = 3 quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Terzo elemento: Conteggio\_azioni3 : < SMS Send an SMS , 4 >

SMS Send an SMS (Conteggio\_azioni3) > SMS Send an SMS (Lista\_numerosità\_minima\_azioni(2))

4 > 2 quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Quarto elemento: Conteggio\_azioni4 : < Twitter Send a direct message , 3 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Conclusione delle iterazioni relative alla lista valore di Trigger-Conteggio\_azioni1.

Proseguo con il successivo elemento presente in Lista\_condizione\_verificata.

Secondo elemento: Trigger-Conteggio\_azioni2

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:

Primo elemento: Conteggio\_azioni1 : < Email Send Email , 3 >

Email Send Email (Conteggio\_azioni1) > Email Send Email (Lista\_numerosità\_minima\_azioni(0))

3 > 2 quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 2 >

IFTTT Send Notification (Conteggio\_azioni2) < IFTTT Send Notification (Lista\_numerosità\_minima\_azioni(1))

2 < 3 quindi Numerosità\_azione di elemento attuale è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, sostituisco il valore nella lista:

Lista\_numerosità\_minima\_azioni(1).value = Conteggio\_azioni2.value;

Terzo elemento: Conteggio\_azioni3 : < PushOver Send High Pry Notifify , 1 >

Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Quarto elemento: Conteggio\_azioni4 : < SMS Send an SMS , 3 >

SMS Send an SMS (Conteggio\_azioni4) > SMS Send an SMS (Lista\_numerosità\_minima\_azioni(3))

3 > 2 quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Conclusione delle iterazioni relative alla lista valore di Trigger-Conteggio\_azioni2

Conclusione delle iterazioni relative alla lista Lista\_condizione\_verificata.

Lista < Conteggio\_azione> Lista\_numerosità\_minima\_azioni:

Email Send Email , 2

**IFTTT Send Notification , 2**

SMS Send an SMS , 2

**Step E-1**

Si calcola min, valore minimo tra quelli presenti in Lista\_numerosità\_minima\_azioni.

$\min = \text{Min} (\text{Lista\_numerosità\_minima\_azioni.value}() )$

$\min = \text{Min} (2, 2, 2)$

$\min = 2$

Si itera su ciascun elemento (Trigger-Conteggio\_azioni) di Lista condizione verificata:

Primo elemento: Trigger-Conteggio\_azioni1

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:

Primo elemento: Conteggio\_azioni1 : < Email Send Email , 6 >

Email Send Email (Conteggio\_azioni1) =

Email Send Email (Conteggio\_azioni1) - min

Email Send Email (Conteggio\_azioni1) = 6 - 2 = 4

Regole\_risparmiate = Regole\_risparmiate + min

Regole\_risparmiate = 0 + 2

Regole\_risparmiate\_utente = Regole\_risparmiate + min

Regole\_risparmiate\_utente = 0 + 2;

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 3 >

IFTTT Send Notification (Conteggio\_azioni2) = 3 > 2 = 1

Regole\_risparmiate = 2 + 2

Regole\_risparmiate\_utente = 2 + 2

Terzo elemento: Conteggio\_azioni3 < SMS Send an SMS , 4 >

SMS Send an SMS (Conteggio\_azioni3) = 4 - 2 = 2

Regole\_risparmiate = 4 + 2

Regole\_risparmiate\_utente = 4 + 2

Quarto elemento: Conteggio\_azioni4 : < Twitter Send a direct message , 3 >

Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Conclusione delle iterazioni relative alla lista valore di Trigger-Conteggio\_azioni1.

Proseguo con il successivo elemento presente in Lista\_condizione\_verificata.

Secondo elemento: Trigger-Conteggio\_azioni2

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:

Primo elemento: Conteggio\_azioni1 : < Email Send Email , 3 >

Email Send Email (Conteggio\_azioni1) = 3 - 2 = 1

Regole\_risparmiate = 6 + 2

Regole\_risparmiate\_utente = 6 + 2

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 2 >

IFTTT Send Notification (Conteggio\_azioni2) = 2 - 2 = 0

Elimino questo elemento dalla lista di appartenenza.

Regole\_risparmiate = 8 + 2

Regole\_risparmiate\_utente = 8 + 2

Terzo elemento: Conteggio\_azioni3 : < PushOver Send High Pry Notify , 1 >

Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Quarto elemento: Conteggio\_azioni4 : < SMS Send an SMS , 3 >

SMS Send an SMS (Conteggio\_azioni4) = 3 - 2 = 1

Regole\_risparmiate = 10 + 2

Regole\_risparmiate\_utente = 10 + 2

Conclusione delle iterazioni relative alla lista valore di Trigger-Conteggio\_azioni2

Conclusione delle iterazioni relative alla lista Lista\_condizione\_verificata.

### ***Step E-2***

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Coppia\_riferimento \_Trigger-Conteggio\_azioni:

Primo elemento: Conteggio\_azioni1 : < Email Send Email , 2 >

Email Send Email (Conteggio\_azioni1) =

Email Send Email (Conteggio\_azioni1) - Email Send Email (Lista\_numerosità\_minima\_azioni(0))

Email Send Email (Conteggio\_azioni1) = 2 - 2 = 0

Elimino questo elemento dalla lista di appartenenza.

Regole\_risparmiate = 12 + 2

Regole\_risparmiate\_utente = 12 + 2

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 3 >

IFTTT Send Notification (Conteggio\_azioni2) = 3 - 2 = 1

Regole\_risparmiate = 14 + 2

Regole\_risparmiate\_utente = 14 + 2

Terzo elemento: Conteggio\_azioni3 : < SMS Send an SMS , 2 >

SMS Send an SMS (Conteggio\_azioni3) = 2 - 2 = 2

Regole\_risparmiate = 16 + 2

Regole\_risparmiate\_utente = 16 + 2

Elimino questo elemento dalla lista di appartenenza.

Regole\_alto\_livello = Regole\_alto\_livello + min;

Regole\_risparmiate = 18 - 2

Regole\_risparmiate\_utente = 18 - 2

### ***Step E-3***

Lista\_Trigger-Conteggio\_azioni,aggiornata risulta essere:

Trigger-Conteggio\_azioni1

- Chiave: Facebook Post by specific user

- Valore:

Conteggio\_azioni1: < Email Send Email , 4 >

Conteggio\_azioni2: < IFTTT Send Notification , 1 >

Conteggio\_azioni3: < SMS Send an SMS , 2 >

Conteggio\_azioni4: < Twitter Send a direct message , 3 >

Trigger-Conteggio\_azioni2

- Chiave: Twitter Tweet by specific user

- Valore:

Conteggio\_azioni1: < Email Send Email , 1 >

Conteggio\_azioni3: < PushOver Send High Priority Notify , 1 >

Conteggio\_azioni4: < SMS Send an SMS , 1 >



Trigger-Conteggio\_azioni3

- Chiave: App.Net New post by specific user

- Valore:

Conteggio\_azioni2: < IFTTT Send Notification , 1 >

Lista\_condizione\_verificata.clear();

Si torna allo Step D-4.

#### ***Step D-4***

Lista\_Trigger-Conteggio\_azioni ordinata utilizzando come chiave di ordinamento decrescente la lunghezza della Lista\_conteggio\_azioni:

Trigger-Conteggio\_azioni1

Trigger-Conteggio\_azioni2

Trigger-Conteggio\_azioni3

Indice\_attuale\_coppia\_considerata = 1 ;

#### ***Step D-5***

Coppia\_riferimento \_Trigger-Conteggio\_azioni = Trigger-Conteggio\_azioni2;

Si scorrono gli elementi di Lista\_Trigger-Conteggio\_azioni :

Primo elemento: Trigger-Conteggio\_azioni1

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento \_Trigger-Conteggio\_azioni:

Email Send Email

PushOver Send High Pry Notify

SMS Send an SMS

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-Conteggio\_azioni1 :

Email Send Email

IFTTT Send Notification

SMS Send an SMS

Twitter Send a direct message

La prima lista non è uguale oppure contenuta nella seconda.

Indice\_attuale\_coppia\_considerata ++;

La lista Lista\_condizione\_verificata è vuota e indice\_attuale\_coppia\_considerata è minore della lunghezza di Lista\_Trigger-Conteggio\_azioni, si torna allo Step D-5.

**Step D-5**

Coppia\_riferimento\_Trigger-Conteggio\_azioni = Trigger-Conteggio\_azioni3;

Primo elemento: Trigger-Conteggio\_azioni1

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento\_Trigger-Conteggio\_azioni:  
IFTTT Send Notification

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-Conteggio\_azioni1 :  
Email Send Email  
IFTTT Send Notification  
SMS Send an SMS  
Twitter Send a direct message

La prima lista è contenuta nella seconda.  
Lista\_condizione\_verificata.Add(Trigger-Conteggio\_azioni1);  
Indice\_attuale\_coppia\_considerata ++;

Secondo elemento: Trigger-Conteggio\_azioni2

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento\_Trigger-Conteggio\_azioni:  
IFTTT Send Notification

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-Conteggio\_azioni2:  
Email Send Email  
PushOver Send High Pry Notify  
SMS Send an SMS

La prima lista non è contenuta nella seconda.  
Indice\_attuale\_coppia\_considerata ++;

Si è terminata l'iterazione sulla lista Lista\_Trigger-Conteggio\_azioni  
La lista Lista\_condizione\_verificata non è vuota proseguo con lo Step D-6.

**Step D-6**

Lista < Conteggio\_azione> Lista\_numerosità\_minima\_azioni:  
IFTTT Send Notification , 1

Si itera su ciascun elemento di Lista\_condizione\_verificata:  
Primo elemento: Trigger-Conteggio\_azioni1

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:  
Primo elemento: Conteggio\_azioni1 : < Email Send Email , 4 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni

Secondo elemento: Conteggio\_azioni2 : < IFTTT Send Notification , 1 >

IFTTT Send Notification (Conteggio\_azioni2) = IFTTT Send Notification (Lista\_numerosità\_minima\_azioni(0))  
 $1 = 1$  quindi Numerosità\_azione di elemento attuale non è minore di Numerosità\_azione dell'elemento presente in Lista\_numerosità\_minima\_azioni, non sostituisco il valore nella lista.

Terzo elemento: Conteggio\_azioni3 : < SMS Send an SMS , 4 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni

Quarto elemento: Conteggio\_azioni4 : < Twitter Send a direct message , 3 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Conclusione delle iterazioni relative alla lista valore di Trigger\_Conteggio\_azioni1.  
Conclusione delle iterazioni relative alla lista Lista\_condizione\_verificata.

Lista < Conteggio\_azione> Lista\_numerosità\_minima\_azioni:  
IFTTT Send Notification , 1

**Step E-1**

$\min = \text{Min}(1)$   
 $\min = 1$

Si itera su ciascun elemento (Trigger-Conteggio\_azioni) di Lista condizione verificata:  
Primo elemento: Trigger-Conteggio\_azioni1

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Trigger-Conteggio\_azioni1:

Primo elemento: Conteggio\_azioni1 - < Email Send Email , 4 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Secondo elemento: Conteggio\_azioni2 - < IFTTT Send Notification , 1 >

IFTTT Send Notification (Conteggio\_azioni2) = 1 - 1 = 0

Regole\_risparmiate = 16 + 1  
Regole\_risparmiate\_utente = 16 + 1

Terzo elemento: Conteggio\_azioni3 : < SMS Send an SMS , 2 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Quarto elemento: Conteggio\_azioni4 : < Twitter Send a direct message , 3 >  
Elemento non presente nella lista Lista\_numerosità\_minima\_azioni.

Conclusione delle iterazioni relative alla lista valore di Trigger-Conteggio\_azioni1.  
Conclusione delle iterazioni relative alla lista Lista\_condizione\_verificata.

### ***Step E-2***

Si itera su ciascun elemento Conteggio\_azione di Lista\_conteggio\_azione, valore della coppia Coppia\_riferimento \_Trigger-Conteggio\_azioni:

Primo elemento: Conteggio\_azioni1 : < IFTTT Send Notification , 1 >

IFTTT Send Notification (Conteggio\_azioni2) = 1 - 1 = 0

Regole\_risparmiate = 17 + 1  
Regole\_risparmiate\_utente = 17 + 1

Regole\_alto\_livello = Regole\_alto\_livello + min;  
Regole\_risparmiate = 19 - 1  
Regole\_risparmiate\_utente = 19 - 1

### ***Step E-3***

Lista\_Trigger-Conteggio\_azioni,aggiornata risulta essere:

Trigger-Conteggio\_azioni1

- Chiave: Facebook Post by specific user

- Valore:

Conteggio\_azioni1: < Email Send Email , 4 >

Conteggio\_azioni2: < IFTTT Send Notification , 1 >

Conteggio\_azioni3: < SMS Send an SMS , 2 >

Conteggio\_azioni4: < Twitter Send a direct message , 3 >

Trigger-Conteggio\_azioni2

- Chiave: Twitter Tweet by specific user

- Valore:

Conteggio\_azioni1: < Email Send Email , 1 >

Conteggio\_azioni3: < PushOver Send High Priority Notify , 1 >

Conteggio\_azioni4: < SMS Send an SMS , 1 >

Trigger-Conteggio\_azioni3

-Chiave: App.Net New post by specific user

-Valore:

Elimino Trigger-Conteggio\_azioni3 da Lista\_Trigger-Conteggio\_azioni.

Lista\_condizione\_verificata.clear();

Si torna allo step D-4.

#### **Step D-4**

Lista\_Trigger-Conteggio\_azioni ordinata utilizzando come chiave di ordinamento decrescente la lunghezza della Lista\_conteggio\_azioni:

Trigger-Conteggio\_azioni1

Trigger-Conteggio\_azioni2

Indice\_attuale\_coppia\_considerata = 1 ;

#### **Step D-5**

Coppia\_riferimento\_Trigger-Conteggio\_azioni = Trigger-Conteggio\_azioni2;

Si scorrono gli elementi di Lista\_Trigger-Conteggio\_azioni :

Primo elemento: Trigger-Conteggio\_azioni1

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per Coppia\_riferimento\_Trigger-Conteggio\_azioni:

Email Send Email  
PushOver Send High Pry Notify  
SMS Send an SMS

Contenuto della lista delle chiavi di Lista\_conteggio\_azioni per l'elemento Trigger-Conteggio\_azioni1 :

Email Send Email  
SMS Send an SMS  
Twitter Send a direct message

La prima lista non è uguale oppure contenuta nella seconda.  
Indice\_attuale\_coppia\_considerata ++;

La lista Lista\_condizione\_verificata è vuota e indice\_attuale\_coppia\_considerata non è minore della lunghezza di Lista\_Trigger-Conteggio\_azioni. Si prosegue allo step11.

#### ***Step E-4***

Il ragionamento prosegue in termini semplificati. Si cercano gruppi di regole in un primo momento aventi medesimo trigger ed azioni appartenenti alla stessa azione di alto livello. In un secondo istante si cercano gruppi di regole aventi trigger appartenenti allo stesso trigger di alto livello e medesima azione. A partire da tali gruppi si calcolano le regole di alto livello per cui è possibile fare il mapping relativamente queste regole di basso livello. Terminati questi calcoli, si torna allo step 6.

#### ***Step D-5***

Sono già stati considerati tutte le coppie presenti nella lista ordinata Lista\_Trigger-Conteggio\_azioni, si torna allo step 2.

#### ***Punto D***

Si ripete l'algoritmo per ogni raggruppamento individuato allo step 1.  
Quando sono stati considerati tutti i raggruppamenti per l'utente attuale, si prosegue allo step 13.

#### ***Punto F***

Non si considerando le liste Lista\_trigger\_senza\_categoria\_alto\_livello e Lista\_azioni\_senza\_categoria\_alto\_livello in questo esempio in quanto si è detto essere vuote.  
Si torna al Punto B e Punto C.

***Step 1***

Il ragionamento prosegue analizzando tutti gli autori del dataset. Una volta terminati gli autori da analizzare si va allo step 14.

***Step conclusivo***

Nel contatore “Regole\_risparmiate” è contenuto il numero di regole che si eviterebbe di creare ragionando con regole di alto livello.

Nel contatore “Regole di alto livello” è contenuto il numero di regole di alto livello che sono state create, le quali comprendono come funzionalità anche le regole terminate nel conteggio regole risparmiate.

L’algoritmo è concluso.

# Capitolo 6

## Analisi dei dati

Gli algoritmi descritti nei capitoli precedenti hanno fornito una grande quantità di dati relativamente al risparmio ottenuto in termini di regole. In questo capitolo sono presentati ed analizzati i dati relativi all'utilizzo di ciascuno dei tre algoritmi. Per i primi due vi è una descrizione rapida, l'attenzione è rivolta in particolare al terzo algoritmo, il quale considera congiuntamente trigger ed azioni di alto livello.

Si tenga conto che i risultati illustrati hanno alla base la scelta di escludere dall'analisi gli autori (e le rispettive regole) che hanno pubblicato un numero di regole pari o inferiore a due. Questo perché per tali autori ragionare per regole di alto livello che permettano di aggregare più regole di basso livello non sembra essere utile. In questo modo le regole analizzate non saranno più 224,590 bensì 115872.

### 6.1 Regole con azioni di alto livello

I dati riportati in questo paragrafo sono relativi all'utilizzo dell'algoritmo precedentemente descritto, facendo però solamente uso delle azioni di alto livello, sono invece considerati i trigger di basso livello. L'obiettivo dell'algoritmo è quello di calcolare il numero di regole in meno che sarebbero state create per ottenere le medesime funzionalità, se si fosse utilizzata una rappresentazione avente azioni di alto livello.

I risultati ottenuti sono i seguenti:

- a Somma del numero di regole considerando tutti gli autori analizzati: 115872
- b Somma del numero di regole rientranti nella suddivisione proposta considerando tutti gli autori analizzati: 104124
- c Somma del numero di regole risparmiate attraverso il raggruppamento in high level action: 13508



- d Percentuale regole comprese nella suddivisione su numero di regole totali considerate (  $b / a$  ): 89,86%
- e Percentuale regole risparmiate su somma regole considerate (  $c / a$  ): 11,66%
- f Percentuale regole risparmiate su somma regole rientranti nella suddivisione proposta (  $c / b$  ): 12,97%

Questi dati mostrano come attraverso l'utilizzo di una rappresentazione che si basa sulle sole regole ad alto livello, si avrebbe un risparmio in termini di regole presenti nel dataset tra l'11% e il 12%.

Tale dato si può leggere come un indicatore che conferma la bontà del ragionamento proposto fino ad ora.

Nel paragrafo 6.3 saranno ripresi questi dati ed utilizzati per approfondire l'analisi dei dati provenienti dall'algoritmo che fa uso di azioni e trigger di alto livello congiuntamente.

## 6.2 Regole con trigger di alto livello

I dati riportati in questo paragrafo sono relativi all'utilizzo dell'algoritmo precedentemente descritto, facendo però solamente uso dei trigger di alto livello, sono invece considerate le azioni di basso livello. L'obiettivo dell'algoritmo è quello di calcolare il numero di regole in meno che sarebbero state create per ottenere le medesime funzionalità, se si fosse utilizzata una rappresentazione avente trigger di alto livello.

I risultati ottenuti sono i seguenti:

- a Somma del numero di regole considerando tutti gli autori analizzati: 106156
- b Somma del numero di regole rientranti nella suddivisione proposta considerando tutti gli autori analizzati: 104624
- c Somma del numero di regole risparmiate attraverso il raggruppamento in high level trigger: 5431
- d Percentuale regole comprese nella suddivisione su numero di regole totali considerate (  $b / a$  ): 98,56%
- e Percentuale regole risparmiate su somma regole considerate (  $c / a$  ): 5,12%
- f Percentuale regole risparmiate su somma regole rientranti nella suddivisione proposta (  $c / b$  ): 5,19%

Questi dati mostrano come attraverso la suddivisione proposta facente uso di trigger ad alto livello si avrebbe un risparmio in termini di regole presenti nel dataset di poco superiore al 5%.

## 6.3 Regole di alto livello

In questo paragrafo saranno presentati i dati come già fatto nei due paragrafi precedenti in modo da avere un confronto immediato relativo al miglioramento ottenuto usando il paradigma di regola ad alto livello.

A seguire sono presentati altri dati, in questo caso ottenuti rendendo più selettivo il criterio relativo al numero di regole pubblicate per gli autori analizzati. Lo scopo di questa analisi è quello di capire meglio l'influenza dell'algoritmo nei casi in cui ha effettivamente senso ragionare in termini di regole più astratte, sintesi di più di una regola di basso livello.

### 6.3.1 Regole di alto livello considerando autori che hanno pubblicato più di due regole

I dati riportati in questo paragrafo sono relativi all'utilizzo dell'algoritmo descritto al punto 5.4, il quale utilizza regole di alto livello. L'obiettivo dell'algoritmo è quello di calcolare il numero di regole in meno che sarebbero state create per ottenere le medesime funzionalità, se si fosse utilizzata una rappresentazione avente regole di alto livello, più astratte.

I risultati ottenuti sono i seguenti:

- a Somma del numero di regole considerando tutti gli autori analizzati: 115872
- b Somma del numero di regole rientranti nella suddivisione proposta considerando tutti gli autori analizzati: 109979
- c Somma del numero di regole risparmiate attraverso il raggruppamento in high level action & trigger: 18489
- d Percentuale regole comprese nella divisione su numero di azioni totali considerati ( b / a ): 94,91%
- e Percentuale regole risparmiate su somma regole considerate ( c / a ): 15,96%
- f Percentuale regole risparmiate su somma regole rientranti nella suddivisione proposta ( c / b ): 16,82%

Questi dati mostrano come attraverso la suddivisione proposta si avrebbe un risparmio in termini di regole presenti nel dataset di poco inferiore al 16%. Questo dato confrontato con i valori relativi all'utilizzo dell'algoritmo con le sole azioni di alto livello (12%) e all'utilizzo dell'algoritmo con i soli trigger di alto livello (5%) mostra immediatamente l'efficacia di combinare assieme i due eventi di alto livello.

Ulteriori approfondimenti sulla bontà di questo ragionamento seguono nel paragrafo successivo.

### **6.3.2 Regole di alto livello considerando autori che hanno pubblicato almeno 10 regole**

I risultati presentati in questo paragrafo prendono in considerazione solamente le regole appartenenti ad autori che ne hanno pubblicate un numero maggiore o uguale a 10. In questo modo i dati risultanti sono in modo più marcato i dati relativi ad autori per cui ragionare in termini di regole di alto livello ed aggregazione di più regole potrebbe essere ragionevole.

Il numero di regole prese in considerazione in questo caso passa da 115872 a 55681. Si analizzano quindi solamente un quarto delle regole totali, in questo modo però il ragionamento è più realistico rispetto all'effettiva applicazione pratica che potrebbe avere.

Il primo dato di interesse è la percentuale di regole risparmiate. In questo caso si ha un risparmio di 13907 regole, il 24,98% delle 55681 considerate. Tale valore mostra un notevole incremento confrontato allo stesso dato considerando però gli autori con un numero di regole maggiore di 2 (15% di regole risparmiate).

Un secondo aspetto di interesse è la dispersione dei risultati analizzati.

Nel primo dei due grafici sotto riportati[Fig6.1], l'asse delle ascisse indica il numero di regole totali per il singolo autore, l'asse delle ordinate il numero di regole risparmiate per il singolo autore.

Il secondo grafico[Fig6.2] invece, raggruppa gli autori sulla base della percentuale di regole risparmiate su quelle totali da esso create.

Da ciascuno dei due grafici colgo un aspetto di interesse.

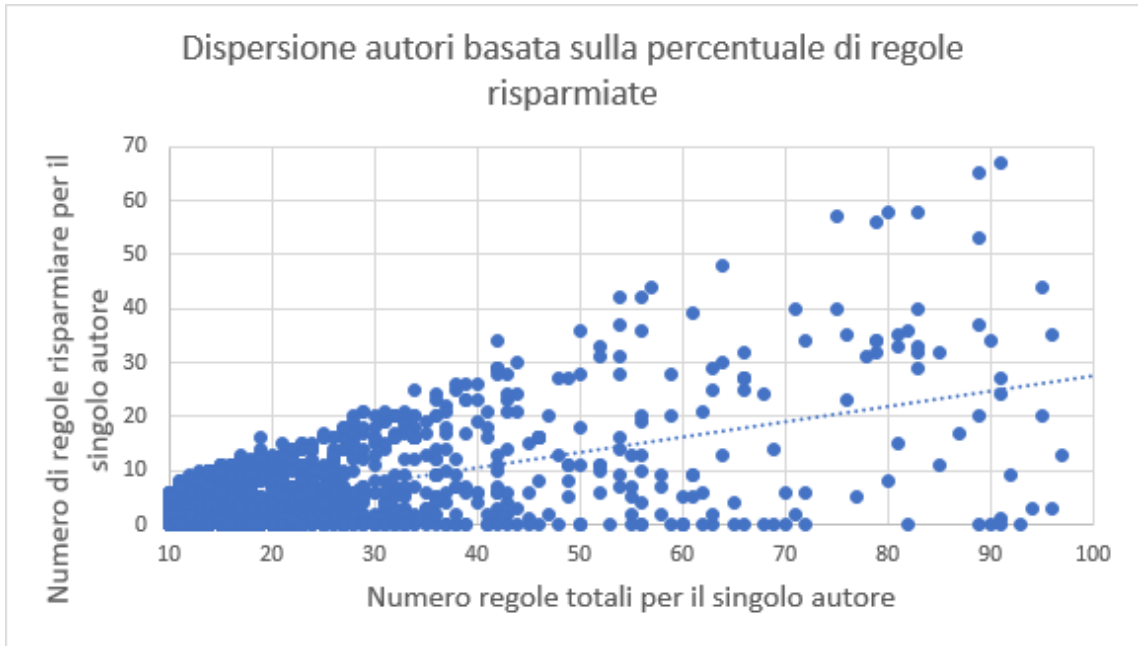


Figura 6.1.

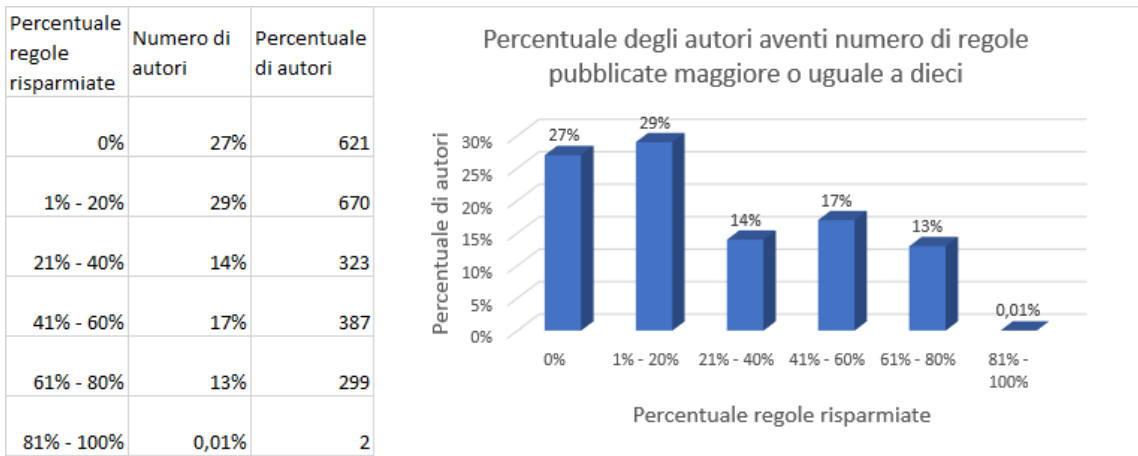


Figura 6.2.

Il grafico relativo la dispersione dei risultati mette in luce l'esistenza di un limite superiore oltre il quale i dati non sono presenti. Si può immaginare una retta che colleghi i punti aventi ordinata maggiore, al crescere del valore sull'asse X. Tale limite (considerato come rapporto tra regole risparmiate e regole totali per il singolo autore) è individuabile tra il 60 ed il 70%. Questo dato è confermato dai valori visibili nel grafico a barre.

Il 13 % degli autori aventi pubblicato 10 o più regole ha un risparmio percentuale compreso tra il 61 e l'80%. Questo valore è molto alto se si considera che effettivamente il numero di autori con un risparmio tra il 70 e l'80% è molto simile all'ultimo dato presente, quello relativo alla percentuale di autori (0.01%) aventi risparmio in termini di numero di regole tra l'81 ed il 100%.

Il grafico a barre evidenzia in modo netto come una grande quantità di autori non ottenga alcun tipo di beneficio dalla suddivisione proposta in termini di trigger ed azioni. Questo è dovuto al fatto che questi autori utilizzino regole in ambiti molto diversi oppure con numerosità di poco superiore a dieci. Anche per essi come per gli autori esclusi dall'analisi, ragionare in termini di paradigma ad alto livello non sembra essere opportuno.

I dati ottenuti applicando l'algoritmo forniscono un ultimo grafico utile. Questo grafico è analogo al precedente come rappresentazione, esclude però un'ulteriore porzione di autori, quelli per cui il risparmio sul totale delle regole pubblicate è pari a zero.

L'obiettivo è quello di ottenere come ultimo valore un dato che tenga in considerazione solamente le persone che sarebbero gli effettivi beneficiari di questo lavoro e non tutti gli utenti di IFTTT che hanno pubblicato regole.

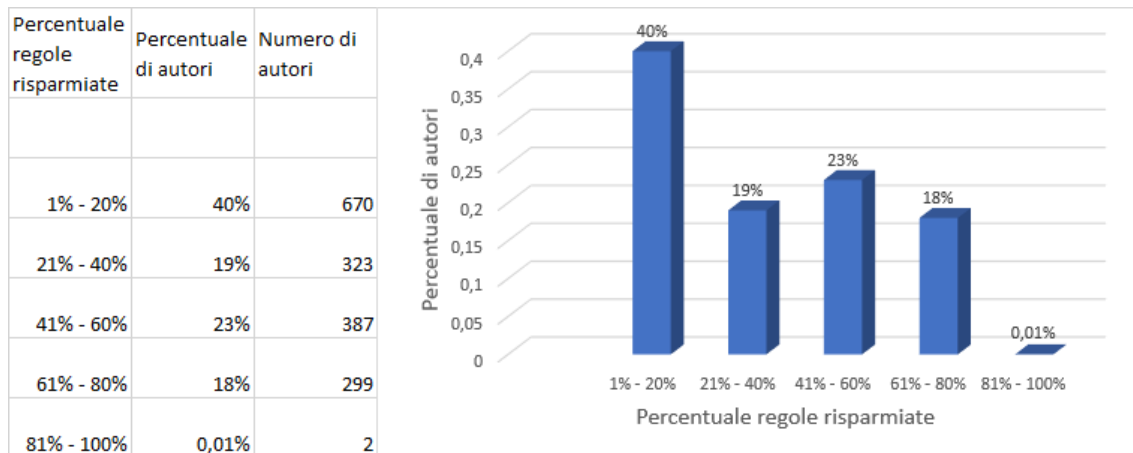


Figura 6.3. Percentuale degli autori aventi numero di regole pubblicate maggiore o uguale a dieci, escludendo quelli il cui numero di regole risparmiate è nullo

Questo grafico considera i 1851 autori (sui 16645 totali del dataset, l'11,12%) che hanno pubblicato un numero di regole maggiore o uguale a 10 e per cui vi è un risparmio percentuale del numero di regole maggiore allo zero per cento. Essi hanno pubblicato 42338 regole (sulle 224590 totali del dataset, il 19%).

Per questi autori in media si avrebbe un risparmio in termini di regole del 33% sul totale di quelle pubblicate. Una regola ogni tre pubblicate sarebbe evitabile. Ciò significa che gli autori che pubblicano molte regole utilizzano applicazioni e servizi affini ed usano tali servizi simili, in modo simile.

Descrivendo in questo modo gli effettivi beneficiari di un ragionamento ad alto livello, ne risulta l'identikit di una persona che si comporta in modo ragionevole, nei cui panni è facile identificarsi. Questa ultima constatazione rafforza la bontà del ragionamento utilizzando un paradigma ad un più alto livello di astrazione rispetto alle regole di basso livello.

## 6.4 Considerazioni

È opportuno in conclusione all'analisi dei dati esprimere due considerazioni aventi come base comune la constatazione che i risultati ottenuti in termini di risparmio siano “il caso peggiore” in quanto non è possibile da quantificare l'influenza di due fattori decisamente rilevanti.

- Nel dataset a disposizione per questo studio sono presenti solamente le regole pubblicate dagli autori. È possibile (certo) che i medesimi autori i quali hanno pubblicato regole ne abbiano create altre di private. Avendo a disposizione l'intero set di regole il dato sul risparmio è probabile che subisca un miglioramento.
- Il secondo aspetto è relativo all'impatto che il ragionamento per regole di alto livello potrebbe portare sugli autori che lo utilizzano. Ragionando ad alto livello non solo si permette di aggregare delle regole di più basso livello, ma sarebbero anche suggeriti alcune applicazioni / servizi i quali sono simili ad uno già utilizzato da un autore ma che ad esso non è venuto in mente di usare. Così gli utilizzatori attraverso un paradigma ad alto livello scoprirebbero servizi magari anche più adatti a svolgere un compito che sono soliti svolgere attraverso altre applicazioni

Questi due aspetti influenzerebbero in modo positivo l'impatto del ragionamento descritto, non è però possibile quantificare oggettivamente quale potrebbe essere questo guadagno.

Nei capitoli seguenti ci si muove nella direzione di questi ultimi due punti analizzati.

Viene prima presentata un'applicazione web sviluppata a scopo di studio, simile ad IFTTT per funzionalità, la quale permette di creare regole con entrambi i paradigmi

(alto e basso livello).

In seguito sono illustrati i risultati ottenuti relativi ad un test utente svolto attraverso questa applicazione.

Il test utente fornirà indicazioni utili per quanto riguarda l'effettivo risparmio di tempo e numero di regole ma anche degli eventuali vantaggi non quantificabili che l'utente ottiene con il paradigma ad alto livello quali ad esempio la scoperta di nuove applicazioni per svolgere determinati compiti o una maggiore soddisfazione nell'uso dell'applicazione.

# Capitolo 7

## Applicazione web per test utente

Allo scopo di valutare l'influenza pratica sulla user experience del paradigma di regole ad alto livello nell'ambito dei servizi pensati per far interagire tra loro applicazioni / elementi connessi di diversi produttori è stato svolto un test utente.

Tale test sarà descritto nel capitolo successivo assieme alla presentazione dei risultati ottenuti. Scopo di questo capitolo è descrivere il funzionamento (ed in breve l'implementazione) dell'applicazione web attraverso la quale il test è stato eseguito dagli utenti che si sono prestati.

### 7.1 Funzionamento

Nell'ambito della creazione di regole, lo scopo del test utente è quello di confrontare una serie di dati relativi ai due approcci:

- paradigma per la creazione di regole di alto livello
- paradigma per la creazione di regole di basso livello

L'applicazione web utilizzata per eseguire il test ha tre funzionalità compiti principali:

- a Permette all'utente scegliere il paradigma a basso livello ed utilizzare questo per la creazione di regole.
- b Permette all'utente scegliere il paradigma ad alto livello ed utilizzare questo per la creazione di regole.
- c Tiene traccia di informazioni relative da una parte alle pagine su cui è stato l'utente, in particolare in merito a pulsanti premuti e tempo di permanenza sulla stessa pagina; dall'altra memorizza le azioni create dall'utente.



### 7.1.1 Utilizzo del paradigma a basso livello

Questa funzionalità rappresenta il modo classico fornito dai servizi per la creazione di regole azione / reazione.

Per l'utente, dopo aver pensato la regola che desidera creare, è possibile selezionare un'applicazione/ servizio per eseguire la regola [Fig7.1] ed in questo scegliere l'evento trigger tra quelli di quel servizio [Fig7.2], utile per realizzare la regola pensata. In modo analogo si comporterà scegliendo l'evento azione. Infine confermerà la creazione della regola. Con la scelta di ciascun evento, sia esso trigger o azione, l'utente potrà specificare le opportune caratteristiche di questo [Fig7.3].

Un esempio:

Selezionando il servizio Office 365 [Fig7.1] ed in esso l'evento azione "Mandami una mail" [Fig7.2], posso impostare alcune opzioni quali l'oggetto della mail, cosa comparirà nel testo [Fig7.3].



Figura 7.1. Applicazione web: Seleziona un canale di azione

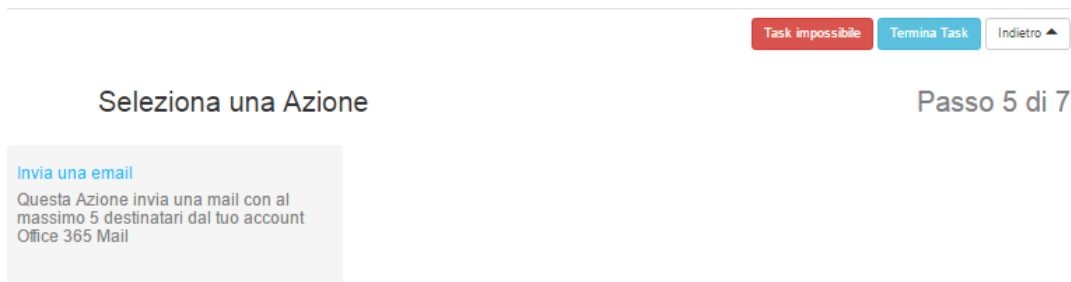


Figura 7.2. Applicazione web: Seleziona una azione di basso livello

The screenshot shows a web application interface. At the top right, there are three buttons: "Task impossibile" (red), "Termina Task" (blue), and "Indietro ▲" (white). Below these, the main heading is "Completa l'Azione" on the left and "Passo 6 di 7" on the right. The form consists of three input fields: "Destinatario", "Oggetto", and "Contenuto". Below the input fields is a blue button labeled "Crea Azione".

Figura 7.3. Applicazione web: Completa l'azione di basso livello

### 7.1.2 Utilizzo del paradigma ad alto livello

Questa funzionalità rappresenta il modo innovativo proposto nella tesi per la creazione di regole azione / reazione.

Per l'utente, dopo aver pensato la regola che desidera creare, è possibile selezionare la categoria di appartenenza di tale regola [Fig7.4] ed in questo scegliere un evento trigger tra quelli presenti nella categoria [Fig7.5], utile per realizzare la regola pensata. In modo analogo si comporterà scegliendo l'evento azione. Infine confermerà la creazione della regola. Con la scelta di ciascun evento, sia esso trigger o azione, l'utente potrà specificare le opportune caratteristiche di questo e la / le applicazioni (tra quelle proposte) che esso intende utilizzare al fine di creare la regola [Fig7.6].

Un esempio:

Selezionando la categoria Social [Fig7.4] ed in essa l'evento azione "Segnalami in privato" [Fig7.5], posso impostare alcune opzioni quali cosa comparirà nel testo da inserire ed inoltre, solo a questo punto, devo scegliere quale/ quali servizi utilizzare, tra quelli proposti, per eseguire l'azione selezionata [Fig7.6].

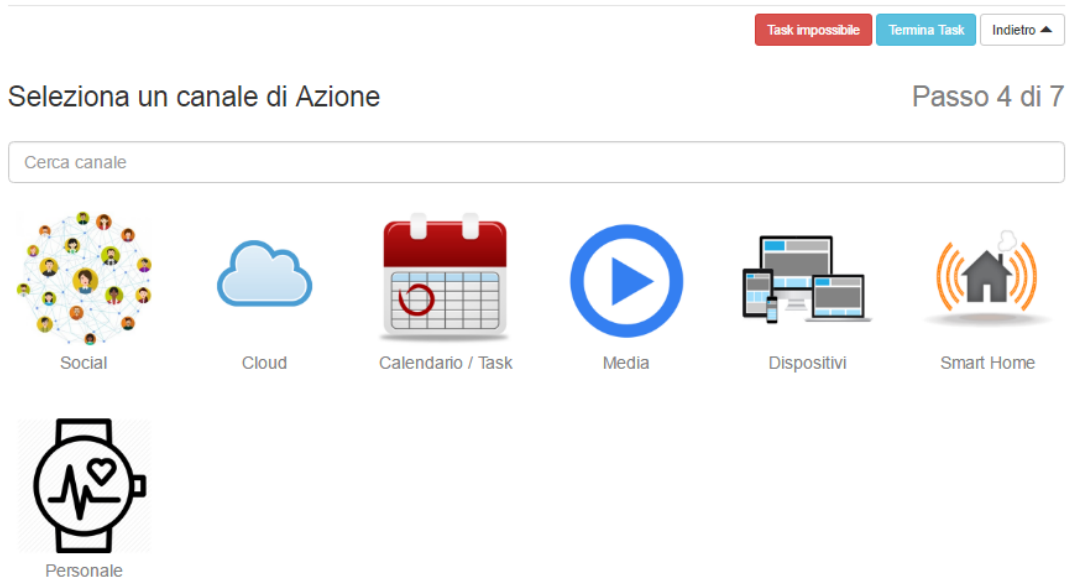


Figura 7.4. Applicazione web: Seleziona una categoria di azione



Figura 7.5. Applicazione web: Seleziona una azione di alto livello



Figura 7.6. Applicazione web: Completa l'azione

### 7.1.3 Memorizzazione informazioni utente

Questa funzionalità si occupa di memorizzare le informazioni relative l'utente che sta eseguendo il test.

In particolare sono memorizzate le informazioni relative:

- i pulsanti premuti dall'utente e l'istante di tempo in cui questo è stato fatto. Attraverso questi dati è possibile capire su quali pagine l'utente si è più soffermato, il tempo totale impiegato per svolgere un task, i passaggi che esso ha fatto (eventuali avanti-indietro) per concludere la creazione della regola.
- le regole (trigger ed azione) create dall'utente. Attraverso questo dato è possibile capire se l'utente è riuscito ad eseguire il task assegnato ed anche confrontare il numero di regole necessarie per svolgerlo con i due paradigmi.

Conclusa questa prima parte relativa le funzionalità dell'applicazione web, nel prossimo paragrafo si illustrano brevemente le tecnologie usate per svilupparla.

## 7.2 Implementazione

Di seguito sono forniti i dettagli tecnici con cui è sviluppata l'applicazione web, senza entrare nel merito del codice che è stato sviluppato.

Il programma elaborato per permettere la raccolta dati attraverso i test utente è una Java Enterprise Edition (JavaEE) applicazione web la quale sfrutta le seguenti tecnologie:

- Database: MySql
- ORM (Mapping tra database ed il mondo ad oggetti): Hibernate
- Frontend: AngularJS
- Backend: Spring

Questa applicazione web è eseguita su un server Apache Tomcat.

Nel capitolo successivo sarà presentato il test sottoposto agli utenti, saranno mostrati i risultati raccolti attraverso il test ed infine sarà illustrata un'analisi che tiene conto dei dati raccolti attraverso l'applicazione web, delle risposte date dagli utenti alle domande presenti nel test ed a annotazione relative a commenti orali da parte degli utenti mentre svolgevano il test.

# Capitolo 8

## Test utente

Obiettivo di questo capitolo è quello di presentare il test, i risultati ottenuti ed infine l'analisi di questi.

Il test svolto e l'analisi sono volti a raccogliere dati e successivamente discutere (attraverso un confronto tra i due paradigmi) relativamente tre aspetti principali:

- Il numero di regole create ed il tempo impiegato per crearle, ottenuto mediante la funzionalità dell'applicazione web che permette di memorizzare i dati relativi all'uso di essa da parte dell'utente
- L'usabilità dell'applicazione web con i due paradigmi, ottenuta mediante i commenti orali, le risposte date alle domande presenti nel test e l'effettiva realizzazione di ciò che era richiesto di fare.
- La user experience nell'utilizzare l'applicazione web, valutata in base ai commenti / pensieri degli utenti durante lo svolgimento del test

Nell'analisi di questi tre aspetti avranno un ruolo importante i commenti fatti dagli utenti durante lo svolgimento del test, in alcuni casi più indicativi di quello che sono i dati raccolti dall'applicazione web oppure la risposta ad una domanda sotto forma di numero valutativo.

### 8.1 Presentazione del test

In questo paragrafo sarà presentata la modalità con cui è fatto il test e successivamente è riportato il test. Infine sono elencate le regole utili a risolvere gli scenari nel modo ottimo con entrambi i paradigmi.

### 8.1.1 Struttura e modalità

Il test che svolgeremo confronterà l'utilizzo delle regole di alto livello e di basso livello avendo come duplice obiettivo da un lato valutare la comprensione dei due paradigmi di regole da parte dell'utente, dall'altro constatare se vi sia o meno un vantaggio in termini di tempo risparmiato e di numero di regole create per gli utilizzatori.

Il test avverrà attraverso un'applicazione web che permette di creare regole con entrambi i paradigmi, utilizzati in modo separato.

Il test sarà svolto fornendo agli utenti tre scenari, sarà chiesto loro di eseguire il medesimo scenario prima utilizzando il paradigma a basso livello e successivamente quello ad alto livello (in metà dei casi l'ordine sarà invertito). Al termine di ogni scenario saranno sottoposte alcune domande all'utente che esegue il test. Usando il medesimo scenario per i due paradigmi la pura comprensione della parte testuale influirà nel medesimo modo sulla creazione delle regole. Gli utenti eseguiranno i task con il seguente schema

Gli utenti selezionati per la simulazione saranno persone in grado di utilizzare

UTENTE 1	Task1_LowLevel/HighLevel	Task2_LowLevel/HighLevel	Task3_LowLevel/HighLevel
UTENTE 2	Task2_LowLevel/HighLevel	Task3_LowLevel/HighLevel	Task1_LowLevel/HighLevel
UTENTE 3	Task3_LowLevel/HighLevel	Task1_LowLevel/HighLevel	Task2_LowLevel/HighLevel
UTENTE 4	Task1_HighLevel/LowLevel	Task2_HighLevel/LowLevel	Task3_HighLevel/LowLevel
UTENTE 5	Task2_HighLevel/LowLevel	Task3_HighLevel/LowLevel	Task1_HighLevel/LowLevel
UTENTE 6	Task3_HighLevel/LowLevel	Task1_HighLevel/LowLevel	Task2_HighLevel/LowLevel
UTENTE 7	Task2_LowLevel/HighLevel	Task1_LowLevel/HighLevel	Task3_LowLevel/HighLevel
UTENTE 8	Task2_HighLevel/LowLevel	Task1_HighLevel/LowLevel	Task3_HighLevel/LowLevel

disinvoltamente le applicazioni web, i servizi e i dispositivi coinvolti nel test. Questi rispecchiano i reali utilizzatori delle applicazioni web basate su regole (come IFTTT) le quali raramente sono fruite da persone poco affini con la tecnologia.

Tra gli otto utenti selezionati sarà presentata un'ulteriore raggruppamento. La metà degli utenti sono persone aventi notevole esperienza nella programmazione, l'altra metà ha esperienza nulla in questo campo.

Gli scenari utilizzati nel test saranno presentati come narrazione di una situazione verosimilmente reale rispetto a quello che è la vita quotidiana di una persona che faccia uso modesto di dispositivi tecnologici.

L'utente dovrà quindi fare lo sforzo di immedesimarsi nella situazione per capire quali regole è opportuno creare avendo come obiettivo quello di automatizzare il più possibile operazioni in cui non è necessaria (ma è solita esserci) l'iterazione tra un'applicazione e un essere umano.



### 8.1.2 Testo del test

E' riportato il testo eseguito dalle presone che si sono prestate per fare il test:

Eeguire ciascuno scenario la prima volta attraverso il paradigma di basso livello, la seconda volta con quello di alto livello

#### TEST

Dati da chiedere all'utente inizialmente:

- Età:
- Sesso: M F
- Professione:
- Fai abitualmente uso del pc' (1 mai - 3 spesso nell'arco di un mese - 5 tutti i giorni)
- Hai mai sentito parlare di IFTTT? Sì No
- In caso di risposta affermativa, lo utilizzi? (1 mai utilizzato - 3 ho impostato una o due regole - 5 quotidianamente)
- Esperienza nella programmazione: (1 mai utilizzato - 3 ho impostato una o due regole - 5 quotidianamente)

Scenario 1:

Marco, un uomo di 43 anni, lavora come giornalista presso la testata di un quotidiano. Marco passa molto tempo leggendo articoli su internet, ha un account Google ed è solito salvare gli articoli più interessanti su Google Drive oppure mandarli tramite Gmail (marco.rossi@gmail.com) ad amici. Nel tempo libero è solito scrivere articoli sul blog che possiede (www.ilmiblog.bBlogger.com) per il quale ha appena aggiunto Luca tra i collaboratori, oppure su quelli di cui è lui stesso collaboratore (www.sonocollaboratore1.wWordPress.com, www.sonocollaboratore2.tTumblr.com e www.sonocollaboratore3.bBlogger.com ).

Per Marco è importante sapere immediatamente se un articolo pubblicato su uno dei blog di cui è collaboratore è stato commentato da un utente, in modo da poter eventualmente rispondere al commento in tempo breve.

Inoltre Marco per un po' di tempo vuole leggere con cura gli articoli di Luca non appena quest'ultimo li pubblica, in modo da assicurarsi che siano in linea con la filosofia del blog.

Domande (si ripetono al termine di ogni scenario):

- a) È stato facile svolgere lo scenario nel primo caso  
(1 non facile - 3 né facile né difficile - 5 decisamente facile)
- b) È stato facile svolgere lo scenario nel secondo caso  
(1 non facile - 3 né facile né difficile - 5 decisamente facile)
- c) Lo scenario presentato era facilmente comprensibile?  
(1 non comprensibili - 3 né facile né difficile - 5 facilmente comprensibili)
- d) Gli obiettivi da raggiungere erano facilmente comprensibili?  
(1 non comprensibili - 3 né facile né difficile - 5 facilmente comprensibili)

Scenario 2:

Sara è una ragazza di 27 anni, abita in Via Rossi a Nichelino. È una maestra sportiva, le piace molto fare attività fisica, per questo motivo si reca a lavoro in bici. Abita in una casa smart nella quale sono presenti le luci WeMo intelligenti, i dispositivi elettrici che usa più spesso sono collegati a una ciabatta Smart Things. La casa di Sara è dotata di antifurto intelligente iSmartAlarm. Sara è solita usare il suo cellulare Android per gestire le attività della sua giornata.

Nella sua vita Sara ha tantissimi impegni e spesso esce di casa velocemente, per lei è importante che le luci ed i dispositivi elettrici siano spenti e che l'allarme sia attivo quando lei si allontana da casa.

Sara è solita organizzarsi in anticipo, ogni sera vorrebbe ricevere una notifica sul cellulare contenente le previsioni meteo per il giorno seguente in modo da sapere se farà brutto tempo (pioggia oppure neve) ed organizzarsi di conseguenza poiché non potrà andare a scuola in bici.

Scenario 3:

Carlo è un Ingegnere affermato, per lavoro è spesso in giro, passa così molto tempo sui social network (Facebook, App.Net, Twitter) durante i viaggi lavorativi, in questo modo riesce a mantenere i contatti con le persone della sua vita. Carlo è smemorato, è solito perciò segnare tutti gli impegni lavorativi nella lista delle “cose da fare” di Todoist e tutti gli impegni personali sulla lista di Trello. Carlo è un grande appassionato di calcio, nonostante i molti impegni lavorativi non si perde mai una partita della sua squadra del cuore, la Juventus.

Carlo è solito pubblicare post e link ad articoli sui vari social network che utilizza. Carlo ha molti contatti, non tutti utilizzano lo stesso social network. Ogni volta che pubblica qualcosa su uno dei suoi social network Carlo vorrebbe che questo automaticamente venisse pubblicato su tutti e tre i social di cui possiede un account.

Essendo un grande tifoso, quando termina una partita oppure quando ci sono delle novità importanti per la sua squadra del cuore Carlo vorrebbe che queste informazioni venissero in modo automatico segnate all'inizio di tutte le sue liste delle “cose da fare” così da ricordarsi di questa notizia e poter informarsi appena ne ha tempo.

### 8.1.3 Regole per svolgere gli scenari nel modo ottimo

Sono elencate le regole per risolvere gli scenari nel modo ottimo con i due paradigmi.

#### Paradigma di basso livello

##### Scenario 1

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
1WordPress	Mio articolo commentato	Gmail	Send me an email
2Tumblr	Mio articolo commentato	Gmail	Send me an email
3Blogger	Mio articolo commentato	Gmail	Send me an email
4Blogger	Articolo pubblicato da	Gmail	Send me an email

Figura 8.1. Scenario 1, basso livello

##### Scenario 2

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
1 Android Posiz.	Lascio un'area	WeMo	Spegni le luci
2 Android Posiz.	Lascio un'area	Smart Things	Spegni ciabatta
3 Android Posiz.	Lascio un'area	iSmart Alarm	Attiva l'allarme
4 Tempo meteo	Domani pioggia debole	Android	Mandami una notifica
5 Tempo meteo	Domani pioggia forte	Android	Mandami una notifica
6 Tempo meteo	Domani neve	Android	Mandami una notifica

Figura 8.2. Scenario 2, basso livello

##### Scenario 3

CANALE TRIGGER	TRIGGER	CANALE AZIONE	AZIONE
1Facebook	Mio nuovo post	Twitter	Nuovo Tweet
2Facebook	Mio nuovo post	App.Net	Nuovo Post
3Twitter	Mio nuovo Tweet	Facebook	Nuovo Post
4Twitter	Mio nuovo Tweet	App.Net	Nuovo Post
5App.Net	Mio nuovo post	Twitter	Nuovo Tweet
6App.Net	Mio nuovo post	Facebook	Nuovo Post
7 Espn	Concluso l'incontro di	Todoist	Crea task
8 Espn	Concluso l'incontro di	Trello	Crea card
9 Espn	Novità importante per	Todoist	Crea task
10Espn	Novità importante per	Trello	Crea Card

Figura 8.3. Scenario 3, basso livello

## Paradigma di alto livello

### Scenario 1

	CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
1T	SOCIAL	Mio articolo commentato	WordPress, Blogger, Tumblr
2T	SOCIAL	Articolo pubblicato da	Blogger
	CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
1A	SOCIAL	Mandami una notifica	Gmail
2A	SOCIAL	Mandami una notifica	Gmail

Figura 8.4. Scenario 1, alto livello

### Scenario 2

	CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
1T	POSIZIONE	Esco da un area	Android Posizione
2T	POSIZIONE	Esco da un area	Android Posizione
3T	POSIZIONE	Esco da un area	Android Posizione
4T	TEMPO	Domani fa:	Tempo meteo
	CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
1A	SMART HOME	Spegni le luci	WeMo
2A	SMART HOME	Spegni ciabatta	Smart Things
3A	SMART HOME	Attiva l'allarme	iSmart Alarm
4A	SOCIAL	Mandami una notifica	Android

Figura 8.5. Scenario 2, alto livello

### Scenario 3

	CATEGORIA TRIGGER	EVENTO TRIGGER	CANALE TRIGGER
1T	SOCIAL	Nuovo social post	Twitter, Facebook, App.Net
2T	INTERESSI	Notifica sportiva	Espn
	CATEGORIA AZIONE	EVENTO AZIONE	CANALE AZIONE
1A	SOCIAL	Nuovo social post	Twitter, Facebook, App.Net
2A	CALENDAR/ TASK	Nuova cosa da fare	Todoist, Trello

Figura 8.6. Scenario 3, alto livello

## 8.2 Presentazione dei risultati

In questo paragrafo saranno presentati i dati raccolti, attraverso i quali nel capitolo successivo si svolgerà la valutazione dei tre aspetti citati nell'introduzione. Anche la presentazione dei dati sarà mostrata seguendo i tre aspetti che saranno analizzati: il numero di regole create ed il tempo impiegato, l'usabilità dell'applicazione web, la user experience dell'utente nell'utilizzare il servizio.

- Il numero di regole create ed il tempo impiegato per crearle

- \* Il numero di regole create:

La situazione ottima per eseguire i tre scenari è stata mostrata al termine del paragrafo precedente. Attraverso questa soluzione sono necessarie 20 regole per svolgere i tre scenari con il paradigma di basso livello, 8 regole per svolgere gli scenari attraverso il paradigma ad alto livello.

Relativamente all'alto livello, 7 utenti hanno utilizzato 8 regole per svolgere i tre scenari. Un utente ha utilizzato 9 regole.

Relativamente al paradigma di basso livello, due utenti hanno utilizzato 20 regole per svolgere i tre scenari. Sei utenti hanno fatto uso di diciotto regole, eseguendo in modo errato una consegna. In particolare, nello scenario avente Sara come soggetto, tutti e sei gli utenti hanno fatto sì che arrivasse una notifica contenente le previsioni complete per il giorno seguente. Era invece richiesto che venisse inviata una notifica con l'obiettivo di segnalare brutto tempo per il giorno successivo.

- \* Il tempo utilizzato:

L'applicazione web tiene traccia del tempo utilizzato dall'inizio alla conclusione dello scenario. Questo dato risulta difficilmente utilizzabile in quanto il test è stato eseguito da persone, le quali hanno eseguito il test iniziando la creazione della regola per poi fermarsi a rileggere il testo, fare domande oppure commenti, svolgere altre azioni scollegate dall'ambito del test (soffiarsi il naso, andare in bagno).

La valutazione del tempo impiegato è proposta attraverso due considerazioni.

- o La prima ottenuta guardando al modo di svolgere il test e relativa la creazione della singola regola.

Con il paradigma di basso livello gli utenti senza troppi problemi selezionavano l'applicazione (questo perché spesso indicata nel testo

dello scenario), per poi soffermarsi qualche istante di tempo in più per trovare l'evento tra quelli proposti per la singola applicazione (spesso sono molti e presentati senza alcuna logica). In fine senza troppi problemi compilavano i campi contenenti i dettagli della regola.

Con il paradigma di alto livello gli utenti selezionavano come prima cosa la categoria. In alcuni casi questa era banale, in altri casi, spesso riguardo gli eventi relativi i blog, trovare quale fosse la categoria non è stato immediato (è stato necessario entrare all'interno di una o due categorie). In un unico caso, relativo all'invio della notifica, quattro utenti hanno avuto bisogno di accedere a più di due categorie per trovare l'evento desiderato. Successivamente selezionare l'evento all'interno della categoria è stato decisamente agevole in quanto il numero di eventi per ciascuna categoria non era elevatissimo. In fine senza troppi problemi gli utenti compilavano i campi contenenti i dettagli dell'evento e selezionavano le applicazioni / i servizi relativi al trigger / azione creato.

- o La seconda considerazione è relativa al numero di regole necessarie per svolgere il medesimo scenario. Sono infatti necessarie più regole di basso livello rispetto a quelle di alto livello. L'azione di creare una regola è svolta un numero maggiore di volte con il paradigma di basso livello, anche il tempo totale per svolgere lo scenario verrà influenzato da questo dato.

- L'usabilità dell'applicazione web:

L'usabilità dell'applicazione web è valutabile utilizzando, come dati numerici e statistici, il numero scenari non eseguiti nel modo corretto e le risposte degli utenti alle domande del test . A questo si aggiungono i commenti orali fatti dagli utenti nello svolgimento del test.

- o Il numero di scenari non eseguito nel modo corretto è zero relativamente il paradigma di alto livello, è sei (su un totale di 24, 3 scenari per 8 utenti) relativamente il paradigma di basso livello. In particolare è sempre lo stesso scenario avente come soggetto Sara, e la stessa regola relativa il meteo ad essere stata eseguita in modo non conforme a quanto scritto nel testo.

Questo dato ha grande valore se considerato assieme a quanto dichiarato dagli utenti. Tutti hanno risposto assegnano il punteggio massimo alle domande relative la comprensione dello scenario e delle richieste, salvo poi creare regole non soddisfacenti a pieno la richiesta, con il paradigma di basso livello.

- o Le risposte alle domande sono presentate nel grafico sottostante [Fig8.7]. La “serie1” rappresenta le risposte relative al paradigma di alto livello, la “serie2” le risposte relative l’utilizzo del paradigma di basso livello. Per

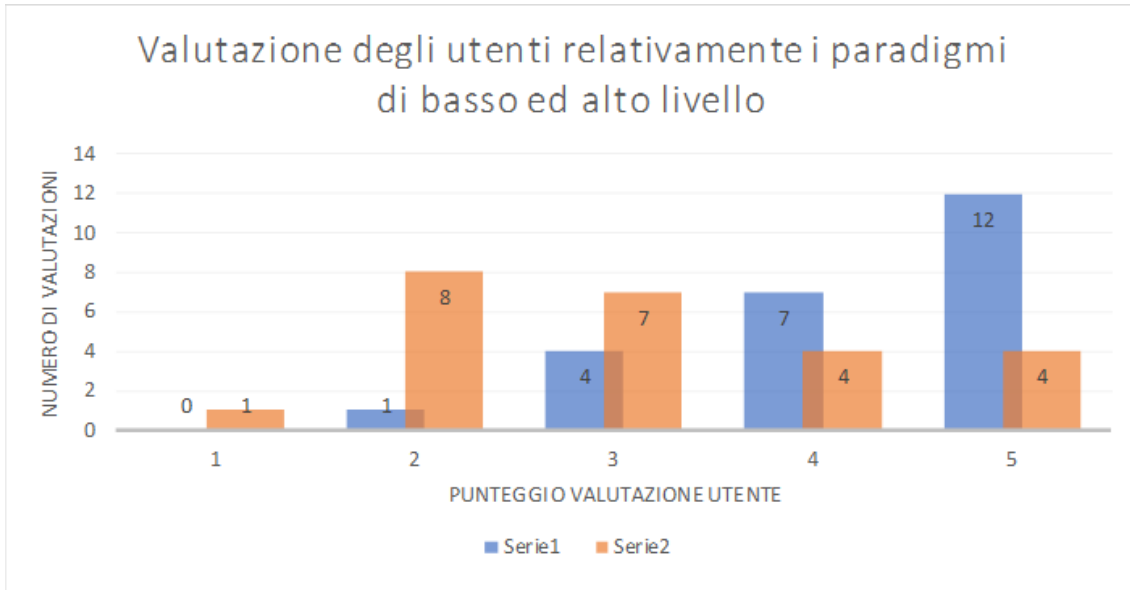


Figura 8.7. Valutazioni facilità nello svolgere il test:  
Confronto tra il paradigma di alto e basso livello

quanto riguarda il basso livello, la maggior parte delle risposte, quindici su venticinque, relative al basso livello sono concentrate tra i punteggi due e tre (né facile né difficile).

Le risposte aventi valutazione cinque sono quattro, di cui tre date dal medesimo utente. Un utente ha attribuito punteggio 1 (non facile) allo svolgimento di uno scenario di basso livello.

Soffermando l’attenzione sull’alto livello (serie1), la maggioranza delle risposte sono concentrate tra i punteggi quattro e cinque (decisamente facile). Su un totale di ventiquattro scenari, gli utenti hanno valutato decisamente facile lo svolgimento attraverso il paradigma di alto livello, in 12 occasioni. Non vi è alcuno scenario a cui è stata assegnato uno (non facile) come valutazione.



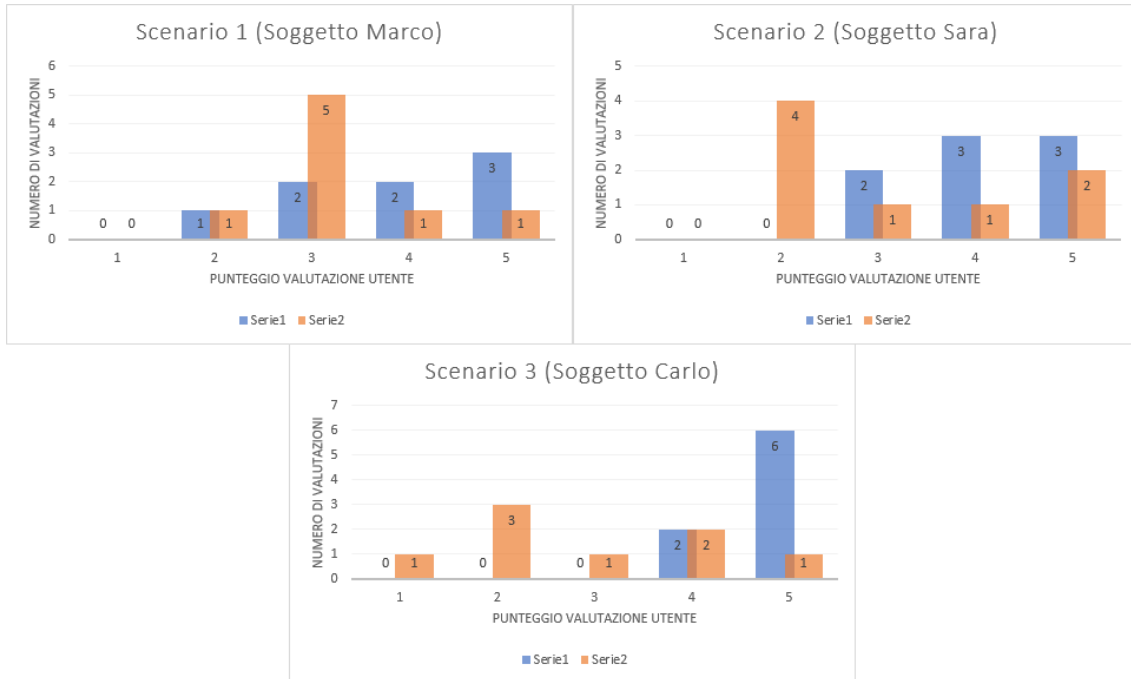


Figura 8.8. Valutazioni facilità nello svolgere i singoli scenari del test:  
Confronto tra il paradigma di alto e basso livello

Anche guardando lo stesso dato considerando ciascuno dei tre scenari singolarmente [Fig8.8], emergono le medesime considerazioni fatte in precedenza. Questo indica come i dati aggregati non sono fortemente condizionati da un certo scenario. Infatti in tutti e tre gli scenari i punteggi, nel complesso, sono maggiori per il paradigma di alto livello.

- o Svolgendo il test utilizzando i due paradigmi, gli utenti hanno fatto alcune considerazioni e commenti.

I commenti e le considerazioni relative al paradigma di basso livello sono stati:

Alcune applicazioni e servizi presentano molti eventi azione o trigger, è difficile individuare in tempo breve quello adatto senza leggerli tutti.

I servizi e le applicazioni sono molte, navigare la pagina con la scroll bar per trovare quella adatta non è agevole. Gli utenti non si sono mai accorti immediatamente di poter cercare attraverso la barra di ricerca.

Stupore e noia relativi gli scenari in cui hanno dovuto 'creare tante volte la stessa regola'. La diversità dei canali in gioco nella regola non era una motivazione così forte da dare la percezione di star creando una regola nuova e diversa dalla precedente.

Con questo paradigma è possibile avere il controllo completo di quello che si fa, scegliendo la singola applicazione/ servizio e la singola regola appartenente al servizio.

In alcuni casi questo paradigma è molto dettagliato, più di quello ad alto livello.

E' difficile trovare un servizio/ applicazione che faccia una determinata funzione se non è nettamente specificato nel testo (i casi a cui si fa riferimento sono l'utilizzo del canale "Android Location" nello scenario 2 ed all'utilizzo del canale "La gazzetta dello sport" nello scenario 3).

I commenti e le considerazioni relative al paradigma di alto livello sono stati:

E' stato facile e veloce nel creare regole che inizialmente sembravano complesse.

E' difficile trovare alcuni eventi, in particolare mandare una notifica. Questa difficoltà c'è la prima volta, poi il ragionamento della suddivisione è compreso ed è immediato trovare gli eventi per via del basso numero di categorie.

E' facile trovare gli eventi perché le categorie in cui guardare non sono molte.

Mi accorgo se non sto creando una regola nel modo in cui vorrei perché non è proposta l'applicazione che ho individuato per eseguire la regola.

Nel caso del risultato finale dello sport, è indifferente il canale da cui questa notizia è attinta. Non posso avere una visuale d'insieme su quali siano le applicazioni ed i servizi, neppure all'interno della singola categoria.

- User Experience

Per valutare la user experience si utilizzano i commenti ed i pensieri a voce alta fatti dagli utenti durante lo svolgimento del test. A questi sono aggiunte delle considerazioni pensate osservando il modo in cui gli utenti hanno eseguito il test.

Commenti ed i pensieri a voce alta fatti dagli utenti:

- o Durante lo svolgimento del test con il paradigma di basso livello.

Incredulità e noia nel dover ripetere regole simili, in alcuni casi per sei volte, compilando ciascuno dei dati ogni volta.

Difficoltà nel trovare un applicazione / servizio non conosciuto che facesse una determinata azione.

Tranquillità nel poter visualizzare tutte le applicazioni e servizi assieme ed essere sicuro di riuscire a completare la regola desiderata in quanto tutte le applicazioni disponibili per realizzarla sono elencate.

- o Durante lo svolgimento del test con il paradigma di alto livello.

E' molto rapido creare regole contenenti tante informazioni.

Le categorie sono poche, è facile orientarsi.

Curiosità guardando le categorie e successivamente provare ad aprirle tutte per guardare quali eventi contenessero.

Difficoltà nel trovare immediatamente alcune eventi. In particolare sei utenti su otto hanno avuto difficoltà ad individuare velocemente l'evento trigger "mandami una notifica", scatenato in risposta ad una determinata azione. Tali difficoltà si sono verificate anche per alcuni utenti (due) nell'individuare gli eventi relativi i post.

Considerazioni pensate osservando il modo in cui gli utenti hanno eseguito il test:

Un utente sosteneva come non ci fosse grande differenza tra un paradigma e l'altro, però il tempo effettivo impiegato per eseguire lo scenario nelle due

modalità era molto differente. Nettamente inferiore nell'alto livello, anche se non ne aveva la percezione mentre eseguiva il test.

Un utente, nello scenario 2, ha risposto essere più facile usare il paradigma di basso livello. Questo utente non ha però creato le regole di basso livello corrette per eseguire quanto richiesto dal testo. In particolare nella richiesta del meteo, ha scelto tra i molti trigger proposti, uno che fosse approssimativamente adatto, senza leggere nel dettaglio se tra i molti trigger ne fosse presente uno più adatto.

Un utente ha creato, nello scenario 3, le sei regole necessarie per connettere i tre social network. All'inizio ha mostrando stupore e successivamente affermando "se il sito mi permette di fare questo, non ci sarà altro modo di farlo". Gli utenti si abituanano rapidamente a ragionare nel modo in cui un servizio lo propone.

Gli scenari sono stati presentati scrivendo esplicitamente quale applicazione dovesse fare cosa. Questo è esattamente il ragionamento che sta sotto il paradigma di basso livello.

In questo paragrafo sono stati presentati in modo imparziale i dati ottenuti attraverso i test utente, evitando per quanto possibile un confronto diretto. Nel paragrafo successivo saranno ripresi i tre aspetti. Per ciascun aspetto, a partire dai dati illustrati, saranno fatti commenti, confronti ed analisi relativi i due paradigmi indicando anche, per ciascun aspetto, quale dei due paradigmi lo valorizzi meglio.

### 8.3 Analisi e commento dei risultati

In questo paragrafo saranno ripresi i tre aspetti considerati in precedenza. A partire dai dati illustrati, per ciascun aspetto saranno fatti commenti, confronti ed analisi relativi i due paradigmi indicando anche, per ciascun aspetto, quale dei due paradigmi lo valorizzi meglio.

- Il numero di regole create ed il tempo impiegato per crearle
  - Il numero di regole create  
Per svolgere lo scenario con attraverso il paradigma di basso livello gli utenti hanno utilizzato in alcuni casi 18 in altri casi 20 regole. Per svolgere il medesimo scenario con il paradigma ad alto livello sono state necessarie 9

regole. Questo dato mostra in modo netto come il paradigma di alto livello porti grandi benefici in termini di numero di regole create. Il risparmio ottenuto in termini di regole risparmiate attraverso tale paradigma è del 50

- Il tempo utilizzato.

Considerazioni relative la creazione di una singola regola:

Il tempo necessario a creare una singola regola è simile. Potrebbe servire più tempo utilizzando il paradigma di alto livello nel caso in cui vi siano dubbi su quale sia la categoria nella quale si trova un determinato evento. Così come potrebbe essere necessario un tempo maggiore utilizzando il paradigma a basso livello nel caso in cui si voglia creare una regola non conoscendo quale applicazione offre il servizio di cui si necessita.

Considerazioni relative lo svolgimento di uno scenario:

Il tempo necessario a svolgere l'intero scenario (nella vita reale questo sarebbe il tempo necessario per raggiungere un obiettivo per l'utente) è sbilanciato a favore del paradigma di alto livello. Nel caso in cui un certo obiettivo preveda la creazione di regole simili, è facile che queste si ottengano da un'unica regola di alto livello. Utilizzando il paradigma di basso livello si creano più regole ed è necessario più tempo (tutti i valori vanno inseriti una volta per ogni canale si intende utilizzare per le regole).

- L'usabilità dell'applicazione web:

Nel paragrafo precedente si è visto come l'usabilità dell'applicazione web sia valutabile utilizzando, come dati numerici e statistici, il numero scenari non eseguiti nel modo corretto e le risposte degli utenti alle domande del test. I commenti e le analisi saranno presentati considerando uno alla volta questi tre indicatori.

- o Valutando il numero di scenari non eseguito nel modo corretto, risulta chiaro come il paradigma di alto livello sia preferibile a quello di basso livello. Non vi sono stati utenti i quali non hanno aderito alle richieste dei vari scenari, utilizzando il paradigma di alto livello. Questo dato assume un valore ancora maggiore se si considera che nel testo degli scenari sono specificati i canali di trigger ed azione da utilizzare. Questo punto di vista è esattamente quello di cui fa uso il paradigma di basso livello. Nonostante questo, alcuni scenari non sono stati svolti, proprio a basso livello, in modo esaustivo. Il dato relativo il numero di scenari non eseguito nel modo corretto fa emergere come il paradigma di alto livello porti vantaggi nella usabilità dell'applicazione web.

- o Analizzando i dati emersi dalla valutazione degli utenti che hanno svolto i test e presentate attraverso i grafici, il dato più netto riguarda la maggior facilità del paradigma di alto livello rispetto a quello di basso livello sia nel complesso che per i tre scenari singolarmente.

Dai grafici è evidente come il paradigma di basso livello sia stato spesso ritenuto meno facile da utilizzare rispetto a quello di alto livello. In particolare il dato relativo al punteggio massimo, cinque, mostra come il paradigma di alto livello in molti casi si sia rivelato assolutamente adatto.

La stessa cosa si può dire solo in quattro casi per quanto riguarda il paradigma di basso livello. Se poi si considera la provenienza di questo valore, il risultato è ancora più netto. Un solo utente ha attribuito per tre volte la valutazione cinque al paradigma di basso livello, questo stesso utente ha attribuito due quattro ed un cinque all'alto livello. In un calcolo statistico che non prende in considerazione i valori peggiori e migliori, i grafici mostrerebbero in modo ancora più netto la maggior facilità di utilizzo dell'applicazione web attraverso il paradigma di alto livello.

- o A partire dai commenti degli utenti, emersi durante lo svolgimento del test, emergono 4 considerazioni relative l'usabilità:

Il paradigma di basso livello ha molte applicazioni/ servizi e ciascuno di esso ha molte azioni. In alcuni casi risulta macchinoso arrivare all'evento desiderato, il paradigma di alto livello ha poche categorie per cui è semplice muoversi da una all'altra.

Il paradigma ad alto livello presenta categorie non sempre chiaramente esplicative di quello che vi è all'interno. Se si considera però che le categorie individuate ed i nomi assegnati possano essere rese più chiare, questo limite di usabilità è migliorabile senza troppo sforzo. Questa problematica non si riscontra a basso livello.

Il vantaggio maggiore portato dal paradigma di alto livello (percepito senza sforzo dagli utenti) è la possibilità di poter raggruppare in una regola unica, azioni simili fatte da applicazioni diverse. Questo aspetto semplifica di molto l'utilizzo dell'applicazione web.

Il vantaggio di usare il paradigma a basso livello è quello di aver la percezione di avere il controllo completo sulle possibili regole che si creano, avendo a disposizione l'intero elenco dei canali di trigger ed azioni.

- User Experience

Nel paragrafo precedente si è visto come la user experience sia valutata utilizzando i commenti ed i pensieri a voce alta fatti dagli utenti durante lo svolgimento del test a cui si aggiungono considerazioni pensate osservando il modo in cui gli utenti hanno eseguito il test.

La User Experience dell'utente non è arricchita dalla possibilità di creare regole alto livello unendo più regole di basso livello. Gli utenti che utilizzavano questo come primo paradigma non vedevano questo aspetto come eccezionale. Al contrario, utilizzando il paradigma di basso livello, dover creare molte regole simili è valutato in modo fortemente noioso e fastidioso.

Utilizzando il paradigma di alto livello, nei casi in cui la corrispondenza evento-categoria non è stata netta, l'utente si sentiva non adatto all'utilizzo dell'applicazione. Questo ha influenza negativa sulla user experience. Questo aspetto è però facilmente migliorabile attribuendo nomi (ed immagini) più esplicative alle categorie proposte. D'altra parte, per quelle regole in cui non vi erano dubbi su quale categoria selezionare (la maggior parte), la scelta di una categoria e poi di un evento descritti attraverso un linguaggio più vicino a quello dell'uomo è risultata decisamente agevole ed intuitiva, in particolare se confrontata alla scelta di eventi con il paradigma di basso livello. In questi infatti l'utente è dovuto entrare nel merito dei differenti nomi attribuiti ad azioni simili utilizzati da applicazioni diverse (ad esempio la medesima azione di 'pubblicare del testo sulla propria pagina' è chiamata "Post" da Facebook e "Tweet" da Twitter).

Due aspetti influenza fortemente la user experience sono portati dal paradigma di alto livello:

- \* la possibilità di creare regole non conoscendo le applicazioni ed i servizi che svolgono le funzionalità di cui si ha bisogno.  
Come si è visto il paradigma di alto livello fa largamente uso di un linguaggio simile a quello utilizzato nella vita quotidiana delle persone. Questo linguaggio unito alla constatazione che la scelta dell'applicazione / servizio per eseguire le regole avviene come ultimo passaggio prima di concludere la creazione della regola, è il punto chiave per far sì che siano create regole per cui non si conoscono applicazioni e servizi adeguati alle proprie esigenze.
- \* la possibilità di scoprire nuovi servizi ed applicazioni che eseguono le stesse azioni di un servizio già utilizzato dall'utente.  
La scelta del servizio / applicazione da utilizzare è l'ultimo passaggio nella creazione della regola. Quando l'utente sceglie il / i canali è già

perfettamente a conoscenza di quale sarà l'evento che verrà svolto. Leggendo l'elenco delle applicazioni tra le quali è possibile scegliere, l'utente scopre nuove applicazioni oppure scopre che applicazioni che già utilizza hanno funzionalità non conosciute ed inaspettate.

Attraverso questi due aspetti il paradigma di alto livello apporta grande valore aggiunto alla user experience dell'utilizzatore.

In questo capitolo è stato presentato il test eseguito da otto utenti. Sono stati presentati i risultati e ne è stata data una chiave di lettura.

L'ultimo capitolo ha come obiettivo quello di proporre un punto di vista che tenga assieme i dati emersi dagli studi illustrati nei primi 7 capitoli ed il test utente appena descritto. Lo scopo è quello di riassumere come la bontà del paradigma ad alto livello sia sostenuta tanto dall'analisi dei dati, i quali mostrano un risparmio di numero di regole create; quanto dai test eseguiti facendo uso concreto del paradigma. Questi dimostrano i vantaggi portati alla usabilità ed alla user experience dal paradigma di regole più astratto.



# Capitolo 9

## Conclusione

Obiettivo di questo capitolo è ripercorrere quanto descritto nelle sezioni precedenti di questo elaborato mettendo in evidenza i passaggi che hanno portato a dimostrare come l'utilizzo di un paradigma di alto livello, nell'ambito della programmazione basata su regole, sia vantaggioso per l'utente in primo luogo relativamente al risparmio del numero di regole create; in secondo luogo in termini di usabilità ed user experience nell'utilizzare una applicazione web implementante tale paradigma.

E' stato analizzato un dataset contenente le regole pubbliche di basso livello dell'applicazione IFTTT, alla ricerca di correlazioni gli eventi di queste regole. Individuate tali correlazioni è stato possibile ipotizzare un numero ragionevole di categorie, a partire dai concetti utilizzati nella vita di tutti i giorni dalle persone, nelle quali fare rientrare i trigger e le azioni del dataset.

Una volta popolate tali categorie è stato possibile iniziare a ragionare in termini di regole di alto livello. È stato sviluppato l'algoritmo che ha permesso di ricavare dal dataset i dati necessari per un confronto tra il paradigma di basso livello e quello di alto livello. In particolare l'algoritmo ha avuto il compito, per ciascun utente, di trasformare l'insieme delle regole di basso livello da esso create, in regole di alto livello e di tenere memoria di quante regole di alto livello siano state create per coprire l'insieme di regole di basso livello create dall'utente.

Sono quindi stati analizzati i dati estratti dall'algoritmo, con lo scopo di calcolare quante regole ciascun utente avrebbe potuto non creare se avesse utilizzato il paradigma di alto livello. I dati ricavati più significativi riguardano gli autori che hanno pubblicato dieci o più regole, ovvero coloro che fanno uso del servizio in modo almeno moderato. Questa categoria di autori utilizzando il paradigma di regole ad alto livello avrebbe creato il 33% di regole in meno rispetto al paradigma di alto livello. Tale dato non è costante per ogni autore, oscilla tra coloro che sarebbero stati più avvantaggiati (avendo un risparmio massimo del 67%) e coloro che non avrebbero alcun beneficio (poco più dello zero per cento).

Ottenuto questo dato numerico attraverso i calcoli svolti dall'algoritmo, che ha creato automaticamente le regole di alto livello a partire da quelle di basso livello, si è dimostrato come teoricamente un paradigma di regole di alto livello sia vantaggioso per chi lo dovesse utilizzare. Al fine di valutare la bontà del paradigma di alto livello guardando il punto di vista dell'utente, è stato studiato e svolto un test utente. Attraverso questo test, il quale chiedeva all'utente di raggiungere lo stesso obiettivo con i due paradigmi di regole, si è voluto studiare e dimostrare l'impatto positivo delle regole di alto livello anche sulla user experience e sulla usabilità di un applicazione che lo implementa. Questo impatto positivo ha alla base la considerazione che le categorie e gli eventi di alto livello utilizzano un linguaggio più simile a quello abitualmente usato dalle persone nella vita di tutti i giorni.

Le valutazioni attribuite al paradigma di alto livello, dalle persone che hanno partecipato al test, sono state nettamente migliori rispetto a quelle di basso livello, anche se non nel 100% dei casi. Avendo a disposizione un valore numerico compreso tra uno e cinque per indicare la facilità (uno, non facile ; cinque, decisamente facile) nello svolgere ciascuno scenario del test, i risultati ottenuti sono a favore del paradigma di alto livello. A questo è stata attribuita una valutazione di cinque relativamente a dodici scenari su ventiquattro totali, mentre per il basso livello solamente quattro scenari hanno ricevuto valutazione cinque. Ed ancora, per quanto riguarda il paradigma di alto livello, zero scenari hanno ricevuto uno come valutazione ed un solo scenario ha ricevuto due. Sono in tutto dieci le risposte indicanti i valori uno e due nel caso del paradigma di basso livello. Inoltre utilizzando il paradigma di basso livello, non tutti gli scenari sono stati completati nel modo corretto.

Questa serie di dati numerici è volta a dimostrare come il paradigma di alto livello produca due grandi vantaggi per l'utilizzatore: il primo in termini puramente pratici, cioè si ha un risparmio di regole create e quindi di tempo. Il secondo relativo all'esperienza ed alla facilità nell'utilizzo di questo paradigma, studiato per utilizzare un linguaggio ed una prospettiva più vicina al modo di pensare e parlare di una persona.

Si ha un ultimo aspetto decisamente importante definibile come “valore aggiunto”. Attraverso il paradigma di alto livello che prevede di selezionare l'applicazione al termine della procedura di creazione della regola, l'utente ha la possibilità di scoprire nuovi servizi e funzionalità inaspettate di applicazioni già in suo possesso.

Il paradigma di alto livello, ideato a partire dal modo di pensare e dal linguaggio delle persone che ne usufruiscono, ha la forza di unire ai vantaggi puramente pratici una maggior valorizzazione e gradevolezza dell'esperienza utente, volta anche a rendere più semplice ed accessibile l'utilizzo del servizio con lo scopo di agevolare la gestione quotidiana dei dispositivi nell'attuale mondo IoT.

# Bibliografia

- [1] Ericsson, <https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf>
- [2] IFTTT, <https://ifttt.com>
- [3] Universal Programming Of Devices, <http://www.upod.io/datasets.html>
- [4] Journal of Visual Languages and Computing, <https://www.journals.elsevier.com/journal-of-visual-languages-and-computing>
- [5] Practical Trigger-Action Programming in the Smart Home, <http://www.blaseur.com/papers/TriggerActionCHI14.pdf>
- [6] Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes , <http://www.blaseur.com/papers/chi16-ifttt.pdf>