L’adattamento dei gateway al framework di interoperabilità domotica

Vittorio Miori (CNR) – Dario Russo (CNR)– Luca Ferrucci (CNR) – Loredana Pillitteri (CNR)

Breve sommario

Lo scopo del documento è quello di descrivere il ruolo che avranno i gateway all'interno della piattaforma SHELL. In particolare sono descritte le caratteristiche e funzionalità che essi dovranno offrire, con soluzioni pratiche e generali per la loro implementazione.

Parole chiave

gateway, interoperabilità, protocolli, integrazione, standard, domotica, dispositivi, comunicazione, struttura

Indice

Introduzione

I continui e rapidi sviluppi nel campo delle tecnologie della comunicazione e dell'ingegneria elettronica, hanno permesso oggi la creazione di computer, sensori e attuatori sempre più miniaturizzati, sofisticati ed efficienti a prezzi di mercato sempre più competitivi.
Questo ha portato i sistemi software a una rapida crescita in termini di dimensioni [1], complessità, eterogeneità e numero di utenti.
La domanda di applicazioni software in grado di sfruttare questo progresso nell'ingegneria elettronica sta superando la nostra capacità di svilupparle, sia in termini di numero che in termini di sofisticazione richiesta.
Il mercato dei dispositivi intelligenti è in crescita e offre applicazioni sempre più interessanti, ma la mancanza di consenso da parte dell'industria sull'uso di standard e protocolli aperti costituisce un ostacolo importante alla loro diffusione. Inoltre, l'aspetto più importante dell'attuale pratica di sviluppo software è la continua preponderanza di approcci ad hoc guidati dalle esigenze dell'industria, dagli interessi commerciali e dalle pressioni del mercato, piuttosto che dai principi scientifici.
Purtroppo, senza la definizione di un meccanismo comune attraverso il quale i dispositivi e le applicazioni possano scambiarsi informazioni, indipendentemente dal loro standard tecnologico, dal loro marchio o dal loro produttore, le soluzioni tecnologiche offerte non raggiungeranno mai il loro pieno potenziale.
Per questi motivi, la ricerca in questo campo deve affrontare sfide quali standard, scalabilità, eterogeneità, linguaggio comune per la descrizione dei servizi, scoperta di servizi specifici per settore, integrazione con i sistemi informatici esistenti, ecc.
Dal punto di vista del consumatore finale, la mancanza di integrazione tra i diversi standard limita la sua libertà. Il consumatore sarà infatti costretto ad acquistare solo i prodotti conformi ad un particolare sistema e ciò potrebbe avvenire a due condizioni: l'acquisizione contemporanea di tutti i dispositivi necessari nell'edificio, o una conoscenza tecnica che consenta all'utente di acquisire dispositivi conformi a quelli già in suo possesso.
Tuttavia, entrambe le condizioni sono difficili da soddisfare perché nella maggior parte dei casi l'ambiente domestico è molto dinamico: la topologia e la rimozione dei suoi elementi cambiano frequentemente e i dispositivi vengono acquisiti in momenti diversi.
Le aspettative dei consumatori finali di oggi sono quelle di poter utilizzare dispositivi da integrare nella loro vita in modo da migliorarla sfruttando la quantità e la varietà di dati generati da tali oggetti [2].
In particolare ,elettrodomestici, telecamere di sorveglianza, sensori di monitoraggio, attuatori, display, veicoli, macchine e così via [3] dovrebbero poter comunicare e condividere informazioni tra loro sfruttando i servizi sempre più di alto livello come la domotica, automazione industriale, ausili medici, assistenza sanitaria mobile, assistenza agli anziani, gestione intelligente dell'energia e delle reti intelligenti, automobilistico, gestione del traffico, e molti altri.

Il *gateway*

In linea generale, un *gateway* è un termine che indica il servizio di inoltro dei pacchetti generati da una rete locale verso una esterna.
Per il suo funzionamento, spesso il *gateway* ha il compito di tradurre i dati dal protocollo usato dalla rete sorgente a quello usato da quella di destinazione, e per questo è necessario che parli entrambi i protocolli fisici delle due reti. Il servizio può essere svolto sia da un hardware dedicato allo scopo o da un software su un dispositivo *general purpose* come un *personal computer (PC)*.
Spesso i *gateway* non si limitano a fornire la funzionalità di base di *routing* (funzionalità che permette, attraverso una tabella di indirizzamento, di indirizzare i pacchetti provenienti dalla rete sorgente verso l'elemento destinatario), ma integrano altri servizi da e verso la rete locale come *proxy DNS* (funzionalità che si prende in carico le richieste di risoluzione degli indirizzi *DNS* da una rete, generalmente locale, e le inoltra a un *Internet Domain Name Server*.
Può anche tenere in memoria i record *DNS* già cercati per futuri accessi più rapidi), *firewall* (funzionalità che impedisce il passaggio di pacchetti secondo determinate condizioni), *NAT* (funzionalità che modifica l'indirizzo dei pacchetti in transito all'interno di una comunicazione in corso tra due o più nodi della rete) etc., che sono appunto servizi di strato di rete più elevato ovvero applicativo.
Un esempio tipico di *gateway* è quello che inoltra tutto il traffico di una rete *LAN* (*Local Area Network*) verso la rete Internet e viceversa (Figura 1).
In casi più complessi, in cui sono presenti diverse sotto-reti da mettere in comunicazione, il *gateway* rappresenta il riferimento per ognuna di esse instradando nella maniera corretta il traffico dati o reindirizzarlo ad altri *gateway*.

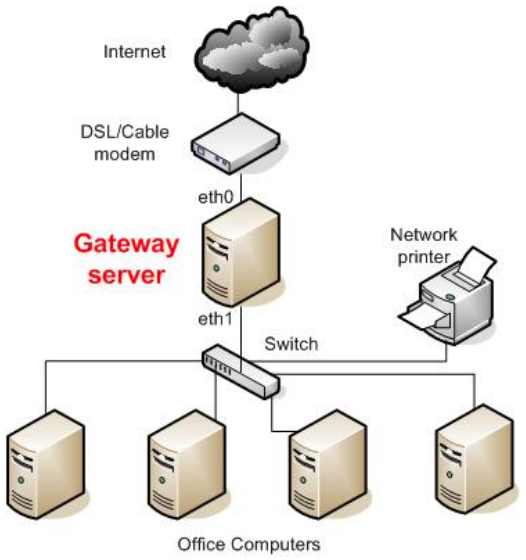

Figura 1: Il gateway che interfaccia una LAN con Internet

Il nome di *gateway*, per estensione, viene dato anche ad un qualsiasi computer o programma che traduca dati da un protocollo ad un altro: ad esempio un computer che provvede a ricevere via E-mail articoli che poi pubblica sui newsgroup è detto un *gateway* da E-mail a news.

Interoperabilità

L'Organizzazione internazionale per la standardizzazione (ISO) ha definito l'interoperabilità [4] come la capacità di due o più sistemi di comprendere, utilizzare le rispettive funzionalità e dare accesso alle rispettive risorse.

Anche la *Healthcare Information and Management Systems Society* [*5*] ha fornito una propria definizione di interoperabilità. L'*HIMSS* lo ha descritto come "la misura in cui sistemi e dispositivi possono scambiare dati e interpretarli". Per essere interoperabili, due sistemi devono essere in grado di scambiare dati e di presentarli successivamente in modo che possano essere compresi da un utente".

Sul piano tecnico, l'interoperabilità contribuisce a ridurre i tempi necessari per un utile scambio di informazioni tra sistemi e tra sistemi ed utenti. Quando i dati sono presentati in modo standard, indipendentemente dalla fonte, è più agevole accedervi rapidamente.

Un esempio pratico di scambio di informazioni in cui il fattore tempo è importante è quello in campo medico. I risultati degli esami del sangue di un paziente fatti la scorsa settimana presso uno studio medico possono essere utilizzati oggi durante un viaggio al pronto soccorso, risparmiando così tempo, costi ed esami non necessari presso l'ospedale.

L'interoperabilità aiuterà gli utenti finali a lavorare in ambienti eterogenei senza dover affrontare la complessità di gestire diverse tecnologie all'interno dell'infrastruttura della loro organizzazione, riducendo il costo degli edifici.

In una visione globale, gli utenti possono rendere disponibili in rete dispositivi appartenenti a tecnologie di più fornitori. Ciò consente di aumentare l'efficienza del flusso di lavoro in qualsiasi ambiente che colleghi oggetti da qualsiasi luogo e con tecnologie diverse.

L'interoperabilità domotica

La domotica (*Home Automation*) è la tecnologia che consente ai dispositivi e ai sistemi intelligenti di automatizzare e integrare il lavoro e le attività domestiche.

Autocontrollo luci, HVAC (riscaldamento, ventilazione e condizionamento), elettrodomestici, serrature di cancelli e porte e altri sistemi in base a determinati obiettivi, è possibile ottimizzare i costi, l'efficienza energetica e comfort, per soddisfare le preferenze degli utenti, per migliorare la sicurezza, la protezione ecc. a casa.

Per gli anziani e i disabili, la domotica può fornire una migliore qualità della vita aumentando la loro autonomia e permettendo loro di essere meno dipendenti da chi si prende cura di loro e dall'assistenza istituzionale.

La popolarità della domotica è aumentata enormemente negli ultimi anni grazie all'accessibilità e alla semplicità molto più elevate offerte dalla connettività di smartphone e tablet.

La domotica può essere considerata il primo dominio di applicazione dell'*IoT* [6] e le sue tecnologie passano sempre più dal semplice controllo e connessione degli apparecchi di un'abitazione isolata ad una connessione e integrazione globale con i servizi disponibili in tutto il mondo.

Le tecnologie domotiche come *KNX*, *UPnP*, *X10*, *LonWorks* e così via, oggi rappresentano una realtà importante che può essere trovata in un gran numero di case. Essi forniscono già un numero elevato e crescente di attività di ricerca nel campo delle applicazioni e dei servizi [7].

L'integrazione del mondo della domotica con quello dell'*IoT* a livello mondiale pone sfide specifiche in materia di ricerca che devono essere affrontate.

Lo scenario di oggi è rappresentato in Figura 2: nel mercato coesistono diversi sistemi tecnologici e dispositivi intelligenti tra loro non interoperabili. Ogni "nuvola" in figura rappresenta uno di essi e può

essere visto come una sottorete domestica all'interno della quale sono presenti sia le infrastrutture hardware e software per il funzionamento, sia i relativi dispositivi conformi.

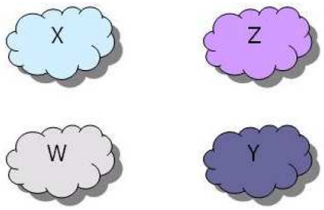

Figura 2: Sistemi come sotto-reti indipendenti

Tutti i dispositivi all'interno della stessa sottorete colloquiano e cooperano per costituire un sistema funzionante ed indipendente, ma chiuso.
Di fatto è impossibile controllare un dispositivo che si trova in una nuvola tramite un altro presente in una nuvola diversa, poiché non esiste nessun tipo di collegamento tra le due sotto-reti.
Per collegamento si intende un'infrastruttura logica che consenta ai dispositivi di qualsiasi sottorete di conoscere tutti i dispositivi presenti nelle altre sotto-reti. Infine, anche qualora fosse possibile avere una visione completa dell'intera rete, sarebbe comunque necessario un meccanismo comune per lo scambio di informazioni tra i dispositivi.
Per far ciò occorre, da un lato riuscire a costruire un'infrastruttura logica di collegamento tra i vari sistemi, dall'altro mettere a punto un meccanismo di comunicazione universale veicolato da quest'infrastruttura.
Una possibile soluzione a questi due problemi è quella di introdurre tra ogni coppia di sottoreti un modulo hardware e/o software che sia provvisto di un'interfaccia verso entrambe le nuvole. Questo significa che ogni coppia di nuvole deve avere un modulo (*gateway*) che deve non solo fornire funzionalità di traduzione tra i due protocolli delle sotto-reti che collega, ma anche conciliare le differenze tra i paradigmi di comunicazione su cui si basano i due sistemi.
Quest'ultima necessità risulta evidente quando i sistemi da collegare presentano caratteristiche notevolmente differenti (ad es. standard a configurazione manuale e *plug and play*).
L'implementazione di questo tipo di *gateway* non può dunque prescindere da una conoscenza approfondita di entrambi i sistemi da collegare. Questa soluzione, tuttavia, risulterebbe poco scalabile a causa del numero di *gateway* necessari al crescere dei sottosistemi da collegare: infatti, all'aumentare del numero delle sotto-reti, il numero di *gateway* da sviluppare cresce sensibilmente.
A questa prima soluzione ce ne è una seconda, che interviene ad un livello superiore che prende ispirazione dalla storia delle reti di calcolatori.
In passato, molti costruttori hanno sviluppato infrastrutture ad hoc per la costituzione di reti che permettessero la comunicazione tra piattaforme e periferiche dello stesso marchio; si pensi ad esempio alla rete *AppleTalk* dedicata ai sistemi *Macintosh*, a quella *NFS* per i sistemi *Unix*, a *NetBEUI* per i PC *Windows* oppure alle reti geografiche *IBM SNA*, *NOVELL/IPX*, ecc.
Queste soluzioni limitavano notevolmente la potenziale espansione delle reti poiché non permettevano la comunicazione tra macchine che supportavano infrastrutture di rete diverse.
Proprio per questo motivo, vi sono stati tentativi da parte di singole aziende, peraltro falliti, di imporre la propria infrastruttura come unico standard, come nel caso di *Microsoft* al rilascio del sistema operativo *Windows 95*.

Di fatto nessuna infrastruttura ha mai raggiunto un'egemonia sulle altre, principalmente a causa dello sviluppo di un'infrastruttura innovativa in grado di superare il vincolo della unicità di piattaforma: questa infrastruttura, di ispirazione *ISO/OSI*, è il noto stack di protocolli *TCP/IP*.
Con lo stesso principio, inserendo una infrastruttura (*framework*) in grado di superare il vincolo di unicità anche nello scenario descritto in Figura 1, sarebbe possibile gestire i diversi sistemi e dispositivi al di là delle differenze tecnologiche. Questo si rende possibile creando, per ogni nuvola, un *gateway* in grado di comunicare da un lato con la sottorete e dall'altro, con la nuova infrastruttura che si pone come intermediario tra le comunicazioni tra due o più sistemi.
Rispetto alla prima soluzione proposta, questa è certamente più scalabile, in quanto richiede solamente un *gateway* per ogni sistema da collegare (Figura 3).

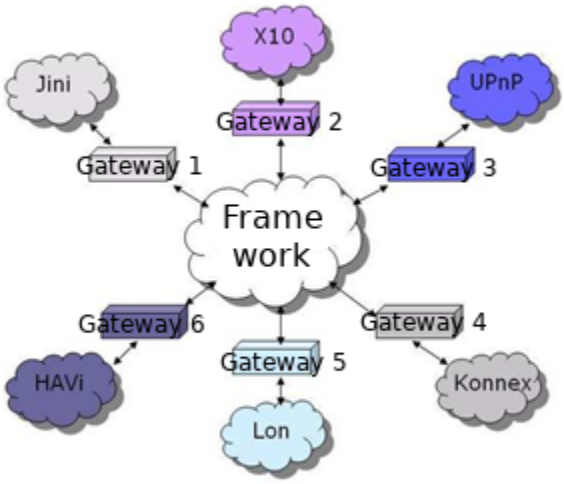

Figura 3: Sistemi come sottoreti interfacciati da Gateway con Framework

Interoperabilità in IoT

Secondo il paradigma dell'*Internet of Things* (*IoT*) [8], gli oggetti sono resi "intelligenti": possono essere scoperti, localizzati, acquisiti, elaborati e scambiati. Le applicazioni *Smart Home & Building* sono particolarmente importanti nello scenario dell'*Internet of Things*, in quanto costituiscono il collegamento tra l'individuo (cittadino, consumatore) e i livelli sovrastanti che implementano il paradigma dell'internet degli oggetti (*Smart City*, *Smart Grid*) [3].

L'*Internet of Things* non è tuttavia solo una visione lontana del futuro, ma è già presente e sta avendo un impatto non solo sugli sviluppi tecnologici. Consente a dispositivi sofisticati di condividere le informazioni direttamente tra loro e con il *cloud* [9], rendendo possibile la raccolta, la registrazione e l'analisi di nuovi flussi di dati in modo più rapido e preciso. In questo modo, è ora possibile sfruttare ad esempio di un'ampia gamma di servizi in grado di migliorare l'autonomia degli anziani [10] e disabili, e di monitorare costantemente il loro stato di salute e individuare eventuali condizioni potenzialmente pericolose [11].

Anche se le definizioni dell'*IoT* nella letteratura scientifica descrivono solo concetti senza specificare formalmente le tecnologie sottostanti e le implementazioni operative, la maggior parte delle soluzioni proposte sono focalizzate su *middleware* in grado di mettere insieme *RFID*, *SOA*, *REST* e protocolli

wireless basati su tecnologie *ZigBee* e *Z-Wave* attraverso la definizione di linguaggi e meccanismi ad hoc per ottenere dati letti dai sensori e sottoporli a servizi *cloud* per la loro elaborazione. Queste soluzioni, tuttavia, sono adatte ad affrontare solo un aspetto del problema nel suo complesso, poiché il mondo dell'internet degli oggetti comprende anche altri importanti sistemi che utilizzano meccanismi diversi, soprattutto negli scenari domotici.

Anche il mondo *IoT* soffre della mancanza di uno standard riconosciuto per l'interoperabilità tra i sistemi. Negli ultimi anni si è registrato un forte sviluppo delle piattaforme dell'*IoT* e la Comunità Europea ha finanziato molti progetti in questo ambito, in particolare nell'ambito del settimo programma quadro. Da questi progetti sono state create diverse soluzioni, ciascuna dedicata alla gestione dell'*IoT* in una specifica area verticale di competenza.

Queste aree di appartenenza sono, ad esempio: *Smart Building*, *Smart Industry*, *Smart City*, *Smart Manufacturing*, *Smart Energy*, *Smart Environment Monitoring*, *Smart Health*, *Smart Living*, *Smart Transport and Mobility*, *Smart Food* / *Water Monitoring*, ecc.

Ciò ha prodotto ottimi risultati, ma purtroppo è sorto un grave problema: le piattaforme sviluppate autonomamente per l'*IoT* non erano interoperabili. In effetti, queste piattaforme sono state pensate ognuna per adattarsi a un determinato scenario, spesso adottando protocolli proprietari di comunicazione, controllo dei dispositivi e delle risorse.

La comunità di ricerca sull'*IoT* ha quindi compiuto un grande sforzo in questa direzione, concentrandosi sullo sviluppo di sistemi in grado di realizzare l'interoperabilità tra soluzioni relative a diversi settori di applicazione, appartenenti a diversi fornitori e che utilizzano diverse tecnologie, per un accesso unificato alle risorse di rilevamento/attuazione.

Nel gennaio 2016 la Commissione Europea ha finanziato con il programma di ricerca *Horizon 2020,* sette progetti che si occupano di interoperabilità nell'*IoT*: *Inter-IoT*, *BIG IoT*, *AGILE*, *symbIoTe*, *TagItSmart*, *VICINITY*, *bIoTope*. In generale, l'approccio tecnico più seguito per affrontare questi problemi è l'implementazione di uno *stack* gerarchico dell'*IoT* che colleghi oggetti intelligenti e *gateway* con il *Cloud*, condividendo le risorse locali disponibili (connettività, computing e storage).

Le soluzioni proposte, in generale, sono basate su architetture che si compongono di 4 livelli:

- *device*: si trovano i dispositivi appartenenti ad una determinata tecnologia;
- *networking*: in cui sono supportati gli oggetti intelligenti, i *gateway* dell'*IoT* e le regole di routing dei messaggi;
- *cloud*: che ospita i dati provenienti dalle piattaforme specifiche, l'analisi dei dati, l'elaborazione dei flussi, la gestione delle piattaforme, ecc. per condividere le risorse tra le piattaforme IoT sottostanti;
- *application*: per consentire la cooperazione e supportare le piattaforme multipiattaforma. A questo livello esistono generalmente ontologie e una rappresentazione semantica comune delle risorse dell'internet degli oggetti per facilitare lo sviluppo di applicazioni specifiche per i vari settori.

L'interoperabilità nell'*IoT* porterà grandi vantaggi che permetteranno l'avvento delle future *smart city* e *smart community*, permettendo la creazione di applicazioni e servizi molto potenti.

Ciò è possibile in quanto l'interoperabilità tra le piattaforme dell'*IoT* consentirà di disporre di dati e servizi unificati, provenienti originariamente da un'ampia gamma di settori di applicazione eterogenei. Anche i dati provenienti dal settore della domotica contribuiranno ad arricchire le banche dati dedicate a questi scopi. A tal fine, dovranno essere risolti i problemi di interoperabilità tra i sistemi di domotica installati all'interno degli edifici.

Il progetto *SHELL*

Partendo da queste premesse, il progetto *SHELL* si pone come obiettivo di studiare e creare un *framework* in grado di trasformare la casa in un ecosistema *IoT* domestico aperto ed interoperabile, composto da dispositivi e tecnologie domotiche eterogenee.

La piattaforma infatti permetterà la comunicazione e l'interazione tra dispositivi tra loro incompatibili, permettendo la creazione di funzionalità avanzate indipendenti dalle tecnologie sottostanti.

La piattaforma *SHELL* vuole essere anche lo strumento abilitante per soluzioni verticali in settori diversi e multifunzionali (energia, sicurezza, comfort), permettendo così di modellare la tecnologia sugli occupanti che accanto alle tradizionali funzionalità domotiche può produrre quelle azioni e sensazioni che lo rendono esattamente a misura di chi lo abita.

Interoperabilità in *SHELL*

L'obiettivo da raggiungere è la creazione di una soluzione che riesca a presentare un unico sistema domotico che nasconda l'eterogeneità di dispositivi e servizi appartenenti ai diversi standard domotici presenti nell'edificio.

Questa soluzione di interoperabilità deve soddisfare i seguenti requisiti specifici di base:

- creazione di *gateway* per ogni diversa tecnologia domotica;
- unificazione di comandi e servizi tra dispositivi appartenenti a standard diversi;
- spazio di indirizzamento unico;
- condivisione dello stato dei dispositivi;
- interazione tra dispositivi in tempo reale;
- interfaccia con sistemi *IoT* esterni all'edificio (ad esempio: *smart city* e *smart communities*).

L'approccio per implementare l'interoperabilità più utilizzato ed emerso negli ultimi anni, si basa su un approccio sintattico. Questo approccio definisce una sorta di "lingua franca" creando un unico linguaggio di alto livello in grado di descrivere in modo uniforme per ogni tecnologia domotica, i dispositivi, le loro funzionalità, i dati, gli eventi e le modalità di interazione. Per funzionare, i *gateway* devono implementare un meccanismo di traduzione che converte i messaggi del sistema domotico (es. eventi, notifiche, comandi) nel linguaggio di alto livello e viceversa.

La Figura 4 mostra un esempio di architettura di un sistema di interoperabilità che fa uso di una "lingua franca".

Quando un dispositivo cambia stato (ad esempio, si preme il pulsante di commutazione), viene generato un *evento(x)* che viene catturato dal *gateway* domotico.

Il *gateway* domotico traduce l'evento in un messaggio *conv_event(x)* in formato "lingua franca".

Il messaggio *conv_event(x)* viene inviato al nucleo del framework di integrazione che elabora e, tramite un sistema di regole, genera una *reazione(x)* corrispondente per attivare l'azione corrispondente su un altro dispositivo (ad esempio per accendere una lampada).

Il framework crea il corrispondente messaggio di alto livello per invocare la funzione del dispositivo interessato e indirizzarlo al corrispondente *gateway* domotico che traduce la *reazione(x)* in un messaggio *conf_react* nella sintassi del linguaggio domotico per la sua esecuzione.

Il vantaggio di questo approccio è che le reazioni sono elaborate in tempo reale e non è necessario alcun database per memorizzare i dati per implementare l'integrazione. Inoltre, il linguaggio medio può permettere di avere un alto livello di flessibilità su come i dati, le funzioni e i comportamenti possono essere condivisi.

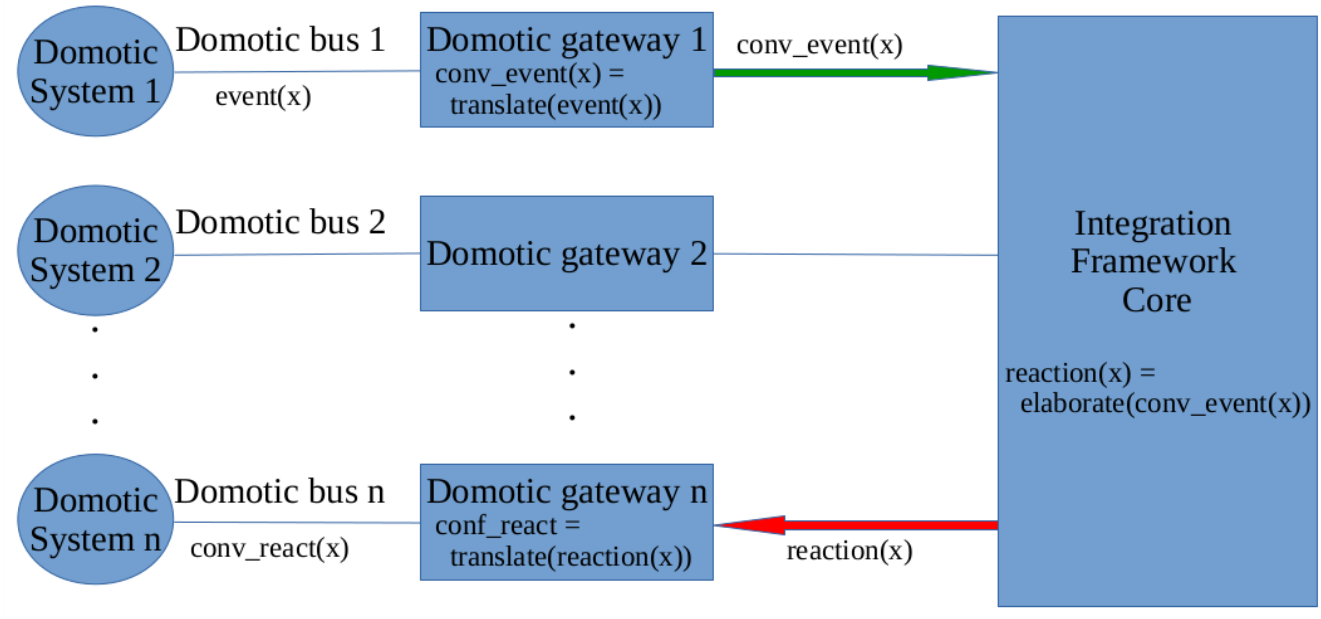

Figura 4: Architettura di un sistema di interoperabilità che usa una "lingua franca"

Considerando che tutti i tentativi di standardizzazione fatti fino a oggi per garantire l'interoperabilità tra dispositivi eterogenei non hanno ancora sortito gli effetti voluti sul mercato, per conseguire gli obiettivi del progetto sarà cruciale affrontare risolvere questo problema.

I *gateway* di *SHELL*

Qualunque sia la soluzione che verrà usata per implementare l'interoperabilità, sarà comunque imprescindibile l'uso di *gateway* per far comunicare il framework con bus domotici, e l'uso di un linguaggio di astrazione interno alla piattaforma.

All'interno della piattaforma *SHELL*, i *gateway* avranno il compito di collegare, interfacciare e integrare funzionalmente i diversi dispositivi e sistemi presenti nell'ambiente (dispositivi reali) alla piattaforma di interoperabilità.

I *gateway* infatti saranno in grado di creare un livello logico di astrazione (dispositivi astratti) per i dispositivi connessi in modo che alla piattaforma di interoperabilità siano nascoste le loro peculiarità hardware e software. Tutti i dispositivi gestiti dai *gateway* saranno visti dal framework come se appartenessero tutti ad un unico grande sistema che comunica attraverso un unico protocollo che è parte integrante di *SHELL*.

Per far ciò, i *gateway* comunicheranno da un lato con il *business logic* del framework utilizzando la "lingua franca" di *SHELL*, e dall'altro con i dispositivi attraverso il protocollo domotico di appartenenza. I *gateway* forniranno le funzionalità e le strutture dati necessarie al fine di tradurre da questo unico protocollo usato in *SHELL*, allo specifico protocollo domotico e viceversa. Considerando che ogni sistema intelligente ha le proprie caratteristiche e modello di funzionamento che la rendono di fatto incompatibile con gli altri standard, occorre un *gateway* per ogni tecnologia domotica (Figura 5).

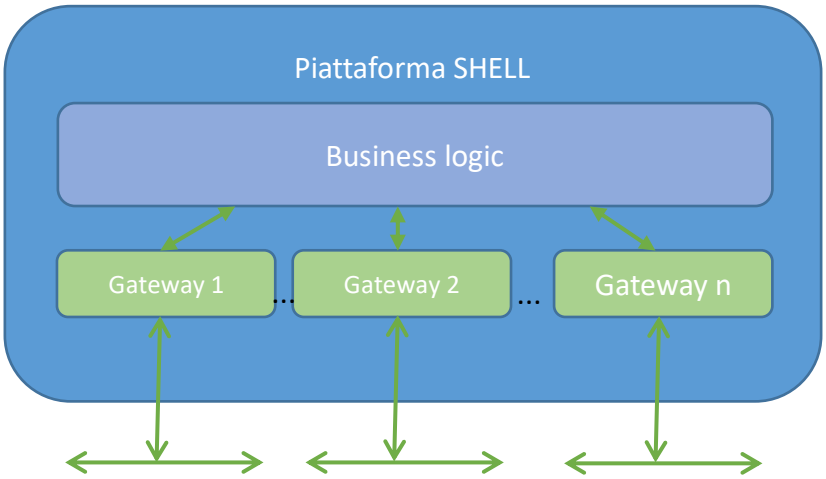

Figura 5: I gateway nell'architettura SHELL

Per implementare queste funzionalità, il ciclo di vita di un gateway sarà:

- inizializzazione del gateway con connessione al framework e al bus del sistema;
- caricamento e astrazione dei dispositivi da gestire;
- fino a che il driver è attivo, ascolto dei comandi in formato *SHELL* da tradurre ed inviare al bus del sistema di pertinenza e gestione della relativa risposta;
- fino a che il driver è attivo, ascolto degli eventi sul bus del sistema da tradurre in formato *SHELL* ed inviare al *business logic* del framework;
- alla chiusura di *SHELL*, disconnessione dal bus del sistema.

Per assolvere ai suoi compiti, ogni gateway di *SHELL* dovrà essere in grado di:

- connettersi e disconnettersi dal canale fisico della rete del sistema domotico;
- individuare i dispositivi del sistema da includere e astrarre;

- creare una corrispondenza tra dispositivi astratti e quelli collegati al bus, comprese le relative funzionalità, comandi, valori e tipi di dato;
- ricevere dal framework comandi in formato *SHELL* provenienti dal *business logic* e convertirli in comandi comprensibili al dispositivo destinatario ed inviare la relativa risposta o un feedback dopo l'esecuzione;
- catturare gli eventi che vengono propagati all'interno della rete del sistema relativi ai dispositivi di interesse e convertirli in eventi in formato *SHELL* per la loro propagazione all'interno del *business logic*.

Connessione al canale fisico della rete del sistema domotico

La domotica permette l'interazione di tutti gli impianti presenti in una casa al fine di migliorarne la gestione e l'ottimizzazione delle risorse. Questo consente a tutti gli elementi che li compongono di dialogare tra loro svolgendo funzioni programmate dall'utente ma anche funzioni svolte in parziale o piena autonomia.

Le tecnologie di automazione, come la domotica, si basano sul *BUS*, un sistema elettrico/elettronico in grado di comandare un insieme integrato di funzioni semplici e complesse.

Fisicamente, il *BUS* può essere un doppino intrecciato che provvede contemporaneamente all'alimentazione e allo scambio di informazioni tra i vari dispositivi, un cavo coassiale o in fibra ottica, ma anche l'alimentazione della rete elettrica oppure l'etere, in radio frequenza o infrarosso (Figura 6). Per controllare un sistema domotico, il *BUS* si compone di elementi che dialogano tra loro, opportunamente connessi a dispositivi con funzione primaria di comando dai quali rilevare informazioni che viaggiano su un supporto di comunicazione.

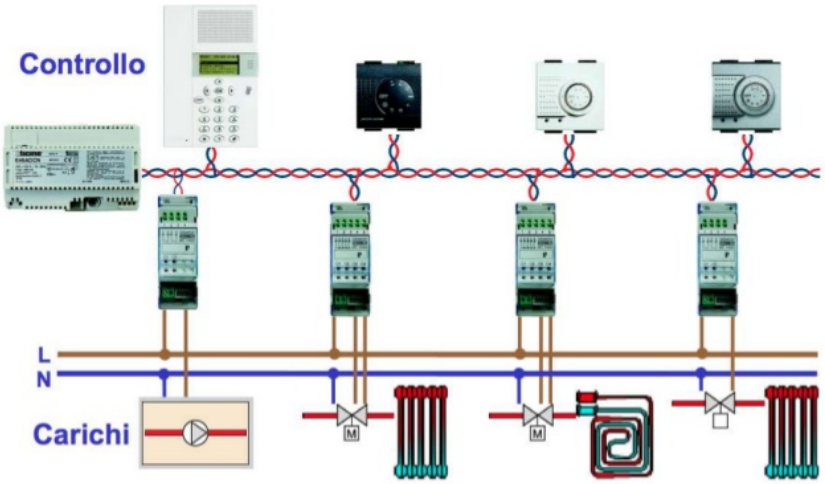

Figura 6: Esempio di impianto basato su BUS

Il supporto di comunicazione è il mezzo fisico mediante il quale i dispositivi si scambiano informazioni. Sul cavo *BUS* i dati vengono trasmessi in formato digitale: una serie di bit codificati secondo uno specifico protocollo di comunicazione.

Chi invia il messaggio è il dispositivo di comando, come ad esempio un pulsante, che dialoga con un dispositivo attuatore, il quale riceve un comando e lo esegue. L'attuatore, una volta ricevuto il messaggio, invia un segnale di ritorno e avvisa della corretta ricezione del comando.

In *SHELL*, la connessione tra il *BUS* e il gateway sarà gestita preferibilmente attraverso una comunicazione di rete su Internet o su una LAN.

La quasi totalità dei protocolli domotici che non sono nativi Internet dispongono di un dispositivo hardware che permette l'incapsulamento dei messaggi domotici in pacchetti *TCP/IP*.

Questo dispositivo è sempre più presente e comune nei sistemi domotici in quanto permette il controllo da remoto dei dispositivi utilizzando applicazioni su telefono e tablet attraverso una connessione Internet.

Questi dispositivi sono chiamati interfacce e svolgono anch'essi funzionalità di gateway tra il bus e la rete *TCP/IP*. Sfruttando questi tipi di dispositivi, ogni gateway *SHELL* potrà connettersi in modo standard al rispettivo *BUS* ed interagire con esso attraverso operazioni di lettura e scrittura.

In Figura 7 è rappresentata una interfaccia *IP* per il sistema *KNX*. In alto si trova la porta per collegare il cavo *Ethernet* mentre in basso, la porta per il *BUS* e la relativa alimentazione ausiliaria.

Figura 7: Interfaccia KNX – IP

Individuazione dei dispositivi del sistema da includere e astrarre;

Per creare i dispositivi astratti all'interno del gateway, occorre in prima istanza individuare i dispositivi con cui si intenderà integrare e che sono collegati alla rete domotica gestita dal gateway. L'individuazione dei dispositivi è strettamente dipendente dal contesto in cui il framework opererà.
In generale infatti, non si può prevedere a priori quali dispositivi saranno presenti nell'ambiente operativo del gateway e per questo deve poter essere il più generale possibile al fine di poterlo applicabile in contesti anche molto diversi tra loro.
Per questo motivo, i dispositivi di un gateway saranno individuati e caricati dinamicamente ad ogni avvio della piattaforma con l'ausilio di file di configurazione esterni al codice eseguibile. Questo permetterà una personalizzazione del gateway in modo flessibile senza dover necessariamente ricompilare il codice per ogni situazione. I file di configurazione conterranno tutte le informazioni necessarie per la rappresentazione dei dispositivi da integrare all'interno della piattaforma *SHELL*.
L'approccio usato per l'importazione delle caratteristiche dei dispositivi domotici al fine di creare quelli astratti nella "lingua franca", è strettamente dipendente dalle funzionalità offerte dal sistema che interfaccia il gateway. Su questo aspetto, è possibile classificare i sistemi domotici in 4 categorie principali:

- *plug-and-play*: i dispostivi si auto presentano quando entrano nella rete domotica. Questi informano della loro presenza e delle funzionalità che offrono (es. *UPnP*). In questo caso il gateway dovrà rimanere in ascolto dei messaggi di presentazione direttamente dalla rete domotica;
- *auto-discover*: il sistema domotico permette di cercare i dispositivi che sono attualmente disponibili e, talvolta, anche di interrogarli sulle funzionalità offerte. In questo caso, il gateway dovrà richiedere al sistema di effettuare una ricognizione dei dispositivi disponibili e di interrogarli per trovarne le funzionalità (es. *BacNet*);
- *file di configurazione*: il sistema domotico è configurato attraverso l'uso di applicazioni esterne in grado di restituire un file di configurazione (es. KNX). In questo caso il gateway dovrà analizzare il file di configurazione per estrarre le informazioni necessarie;
- *nessuno*: il sistema domotico non dispone di alcun meccanismo per l'individuazione dei dispositivi presenti sulla sua rete (es. X10). In questo caso il gateway dovrà far uso di appositi file di configurazione scritti dall'utente.

Creazione di una corrispondenza tra dispositivi astratti e quelli collegati al bus;

Una volta individuati i dispositivi da usare, questi dovranno essere rappresentati all'interno della piattaforma software. Al fine di andare verso una soluzione semantica del framework, la "lingua franca" che verrà usata in *SHELL* dovrà definire una classificazione gerarchica della tipologia di dispositivi supportati dalla piattaforma. In questo modo sarà possibile uniformare e standardizzare i dispositivi astratti *SHELL*, le loro funzionalità e peculiarità.
A questo scopo, per ogni tipologia di dispositivo definito nella classificazione, saranno associate le rispettive funzionalità e i rispettivi comandi predefiniti da *SHELL*.
Le funzionalità in *SHELL* saranno delle categorie che raggrupperanno comandi, mentre i comandi saranno le azioni che un dispositivo sarà in grado di eseguire. Un esempio di funzionalità è "illuminazione" che al suo interno avrà i comandi "accendi luce" e "spegni luce".
Per ogni comando saranno anche definiti i parametri di ingresso (input) e risposta (output), i relativi tipi di dato (data-type) con i relativi range di valori accettati.

Oltre a classificare i dispositivi, funzionalità e comandi, la classificazione definirà anche gli eventi per ogni tipologia di dispositivo. Per evento si intende un cambiamento di stato di un dispositivo appartenente alla rete domotica, dovuto a un fattore sincrono (esecuzione di un comando) o asincrono (in modo spontaneo). Esempi di eventi sono l'accensione di una lampadina, la rilevazione di una temperatura ecc.

Un esempio di classificazione dei dispositivi domotici e delle relative funzionalità è dato dalla ontologia *DogOnt* [12] (Figura 8). *DogOnt* è un linguaggio di modellazione per gli *Intelligent Domotic Environment* (*IDE*), basato sulle tecnologie del Web Semantico.

Tra le funzionalità a offerte da DogOnt, c'è una classificazione gerarchica dei possibili dispositivi domotici, delle relative funzionalità. In figura è mostrata anche una *SimpleLamp* ed è descritta come una lampada che può solo essere accesa o spenta (Figura 9).

Possiede le funzionalità *OnOffFuncionality* e *OnOffNotificationFuncionality* (Figura 10). Possiede i comandi *OnCommand* e *OffCommand* che prendono rispettivamente i valori *on* e *off* (Figura 11). Inoltre possiede uno stato di nome *OnOffState* che prende i valori *on* e *off* (Figura 12).

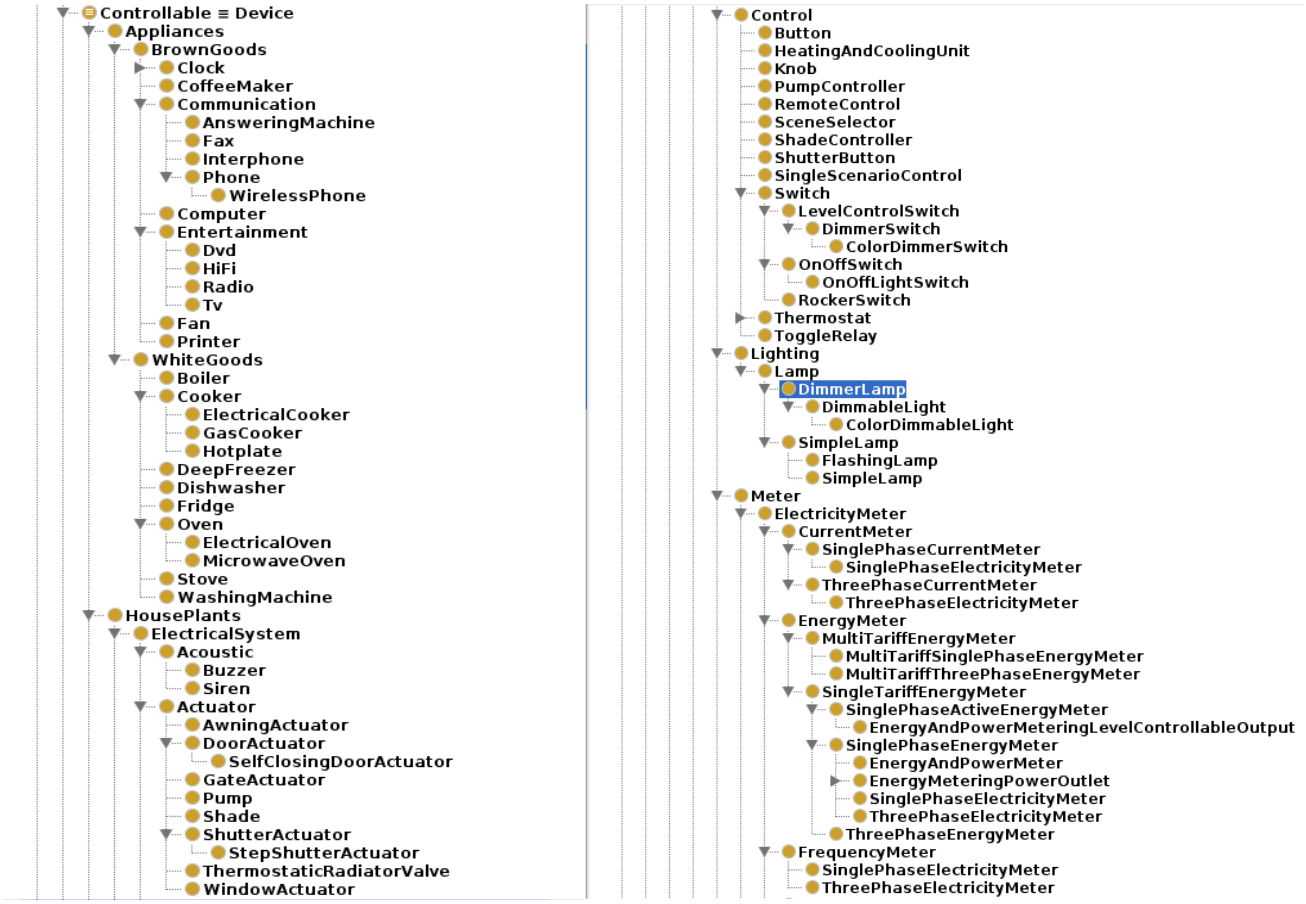

Figura 8: Estratto della classificazione usata in DogOnt

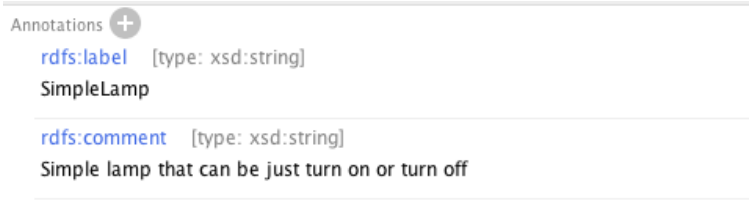

Figura 9: Descrizione della SimpleLight in DogOnt

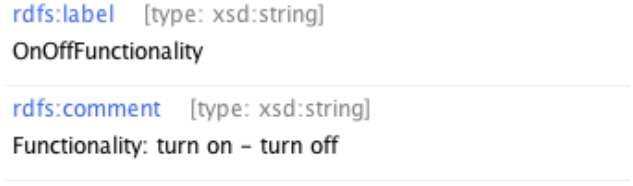

Figura 10: Funzionalità della SimpleLight in DogOnt

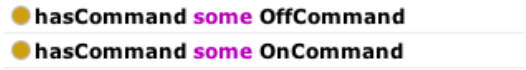

Figura 11: Comandi della SimpleLight in DogOnt

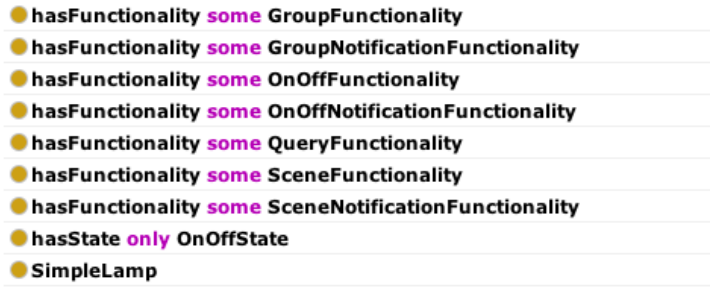

Figura 12: Stati della SimpleLight in DogOnt

Creati i dispositivi, questi dovranno essere caricati tramite dei file di configurazione dal gateway che sarà in grado di crearne una rappresentazione astratta interna.

In caso di uso di un linguaggio di programmazione ad oggetti per lo sviluppo del framework, la rappresentazione potrà avvenire tramite la creazione di una istanza (oggetto) di una classe predefinita allo scopo. La classe da istanziare dovrà contenere tutti i meccanismi base che permetteranno la gestione del ciclo di vita di un dispositivo astratto, compresi i costrutti per la sua creazione, gestione dei messaggi in entrata ed uscita, la comunicazione verso il rispettivo gateway e verso il dispositivo reale per la loro invocazione, la gestione degli stati che può assumere.

Una specifica istanza sarà costruita e personalizzata secondo le informazioni contenute nei rispettivi file di configurazione. Ogni file di configurazione dovrà contenere almeno le seguenti informazioni:

- nome del gateway che prende in carico il dispositivo: mettendo tutti i file in un'unica directory, questo permetterà di distinguere a quale gateway appartiene;
- identificatore univoco del dispositivo all'interno della piattaforma *SHELL*: rappresenterà l'identificativo del dispositivo astratto;
- il tipo di dispositivo: al fine di strutturare una semantica, i dispositivi dovranno avere una classificazione per tipologia;
- la rappresentazione di una struttura per associare un comando *SHELL* relativo al dispositivo descritto, ad un comando proprio della tecnologia domotica, con i rispettivi parametri di input e output;
- la rappresentazione di una struttura per associare i tipi di dato definiti in *SHELL* con quelli definiti dalla tecnologia domotica;
- la rappresentazione di una struttura per associare gli eventi della tecnologia domotica in eventi *SHELL*.
-

Ricevere dal framework comandi in formato *SHELL*

Affinché i cicli di vita del dispositivo reale e del dispositivo astratto siano sincronizzati ed in grado di comunicare tra loro, occorrerà prevedere un meccanismo in grado di associarli tra loro.
Sfruttando il fatto che i dispositivi astratti in *SHELL* avranno un identificatore univoco e che generalmente anche le tecnologie domotiche prevedono un meccanismo analogo per l'identificazione

dei dispositivi all'interno della loro rete, l'associazione potrà essere realizzata creando una struttura dati "a mappa" che permetterà di mettere in relazione l'identificativo del dispositivo astratto con quello reale e viceversa. In questo modo, sarà possibile identificare il dispositivo fisico coinvolto in un messaggio di tipo *SHELL* ed il rispettivo dispositivo astratto.
Definita la tassonomia delle funzionalità e comandi supportati dalla piattaforma *SHELL*, occorrerà creare una struttura dati "a mappa" in grado di mettere in associazione una funzionalità e comando *SHELL* con una funzionalità e comando della tecnologia domotica.
Non tutte le tecnologie domotiche hanno un supporto alle funzionalità come saranno definite in *SHELL*, ma si limitano alla dichiarazione dei comandi.
Visto che i nomi dei comandi dei dispositivi reali all'interno di una tecnologia domotica potrebbero differire secondo specifiche le installazioni (in *KNX*, ad esempio, i comandi *SHELL* possono essere associati a degli indirizzi di gruppo, i quali sono diversi da installazione a installazione) e che per ogni tipo di dispositivo astratto definito in *SHELL* sono già definite e note le funzionalità ed i comandi supportati, nei file di configurazione dei dispositivi sarà possibile fare un'associazione tra il comando *SHELL* ed comando corrispondente della tecnologia domotica.
Allo stesso modo, potranno essere associati anche i tipi di dato e gli eventuali parametri di input e output. Noti i tipi di dato, il gateway dovrà fornire il supporto per una eventuale traduzione tra valori di un tipo di dato *SHELL* e valori di un tipo di dato della tecnologia domotica.
Una volta individuato il corrispettivo dispositivo destinatario, il comando e valori nella tecnologia domotica, sarà possibile effettuare una operazione di scrittura sul bus al fine di permetterne l'esecuzione. Se previsto, con una operazione inversa, sarà possibile ricevere dal bus domotico la risposta all'invocazione del comando e tradurla in un messaggio *SHELL*.

Catturare gli eventi che vengono propagati all'interno della rete domotica

I dispositivi della tecnologia domotica connessi al gateway continueranno il loro ciclo di vita indipendentemente e in modo asincrono dalla piattaforma *SHELL*.
Questo significa che i dispositivi, se previsto dalla tecnologia domotica, continueranno ad inviare periodicamente o a seguito di un cambiamento, il proprio stato sotto forma di valori attraverso un messaggio sul *BUS*.
Questo tipo di messaggi sono chiamati eventi. Gli eventi supportati da *SHELL* saranno definiti secondo una classificazione per tipologia di dispositivo. Per gestire gli eventi, il gateway dovrà rimanere in ascolto costantemente sul *BUS* domotico.
Quando il gateway individuerà un evento sul *BUS*, di questo verranno individuati l'identificatore del dispositivo emettitore e il tipo di evento nella tecnologia domotica.
Attraverso una operazione di mapping, il gateway sarà in grado di individuare il rispettivo dispositivo astratto e il tipo di evento *SHELL*. L'evento *SHELL* individuato sarà notificato al *business logic*, ed in particolare al *Rule Manager*, che provvederà ad intraprendere le operazioni del caso come attuare l'interoperabilità attraverso la generazione di un nuovo comando verso un altro dispositivo di una qualunque tecnologia domotica.
In Figura 13 è possibile vedere un esempio di evento con la conseguente generazione di comando. In questo esempio, la tapparella della tecnologia n si chiude ed invia la notifica sul rispettivo *BUS*.
Tale evento è catturato dal gateway n, tradotto nel linguaggio *SHELL* e notificato al *Rule Manager*. All'interno del *Rule Manager*, una regola definisce che quando si verificherà l'evento della tapparella

che si abbassa, la luce della stanza deve essere accesa. Così il *Rule Manager* genererà il comando di accensione per la lampadina della tecnologia domotica 1.

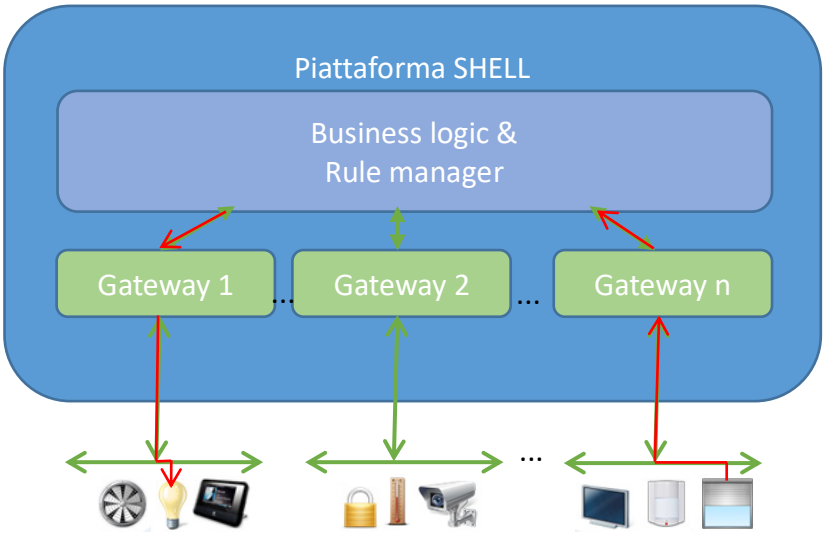

Figura 13: Esempio di cattura di un evento ed attuazione di un dispositivo

Riferimenti

[1] Medvidovic, N., Gamble, R., & Rosenblum, D. (2000, June). Towards Software Multioperability: Bridging Heterogeneous Software Interoperability Platforms. In Proceedings, Fourth International Software Architecture Workshop.

[2] Modre-Osprian, R., Drobics, M., Hayn, D., & Dohr, G. S. A. (2010). The Internet of Things for Ambient Assisted Living.," 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, 2010, pp. 804-809, doi: 10.1109/ITNG.2010.104

[3] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660. doi: 10.1016/j.future.2013.01.010

[4] ISO 14258:1998 (Industrial automation systems -- Concepts and rules for enterprise models)

[5] HIMSS. Definition of Interoperability. Approved by the HIMSS Bard of directors April 5, 2013. http://www.himss.org/sites/himssorg/files/FileDownloads/HIMSS Interoperability Definition FINAL.pdf

[6] Toschi, G. M., Campos, L. B., & Cugnasca, C. E. (2017). Home automation networks: A survey. Computer Standards & Interfaces, 50, 42-54. doi: 10.1016/j.csi.2016.08.008

[7] Miori, V., Russo, D., & Concordia, C. (2012). Meeting people's needs in a fully interoperable domotic environment. Sensors, 12(6), 6802-6824. doi: 10.3390/s120606802

[8] Mukhopadhyay S.C., Suryadevara N.K. (2014) Internet of Things: Challenges and Opportunities. In: Mukhopadhyay S. (eds) Internet of Things. Smart Sensors, Measurement and Instrumentation, vol 9. Springer, Cham. doi: 10.1007/978-3-319-04223-7_1

[9] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616. doi: 10.1016/j.future.2008.12.001

[10] Christophorou, C., Kleanthous, S., Georgiadis, D., Cereghetti, D. M., Andreou, P., Wings, C., ... & Samaras, G. (2016). ICT services for active ageing and independent living: identification and assessment. Healthcare technology letters, 3(3), 159

[11] Miori, V., & Russo, D. (2015). Preventing Health Emergencies in An Unobtrusive Way. In Internet of Things. User-Centric IoT (pp. 242-247). Springer, Cham. doi: 10.1007/978-3-319-19656-5_35

[12] Bonino, D., & Corno, F. (2008, October). Dogont-ontology modeling for intelligent domotic environments. In International Semantic Web Conference (pp. 790-803). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-88564-1_51

Indice delle figure