

SISTEMI MULTIAGENTE PER L'ASSISTENZA SANITARIA DOMESTICA: A CASE STUDY

Danilo Cistaro, Emanuela Malizia

RT-ICAR-CS-18-02

Giugno 2018



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)

– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it

– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.na.icar.cnr.it

– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.pa.icar.cnr.it

1	INTRODUZIONE	3
2.	TECNOLOGIE ABILITANTI AGENT-BASED	6
2.1	MINIATURIZZAZIONE DELL’HARDWARE ED EFFICIENZA ENERGETICA.....	6
2.2	TECNOLOGIE DI RETE	6
2.3	COMPUTER SCIENCE	7
2.4	PIATTAFORME SMART O CYBERPHYSICAL	8
2.5	SISTEMI MULTI-AGENTE	9
3	SISTEMI IOT PER APPLICAZIONI HEALTH E WELLNESS	13
3.1	ARCHITETTURA DI BASE	13
A.	AMON	14
B.	WAITER	15
C.	SISTEMA AD AGENTI DI K. MILLER E S. SANKARANARAYANAN	16
3.2	ANALISI DEI REQUISITI DEL SISTEMA.....	16
3.2.1	<i>Strumentazione</i>	17
3.3	FUNZIONALITÀ DEL SISTEMA	18
4.	CASE STUDY: UN SISTEMA MULTIAGENTE PERVASIVO PER L’ASSISTENZA SANITARIA DOMESTICA	19
4.1	SIMULAZIONE DI SENSORI.....	19
4.2	APPLICAZIONE DEL PAZIENTE.....	19
4.3	APPLICAZIONE DEL MEDICO	21
4.4	APPLICAZIONE DELL’OPERATORE SANITARIO SUL TERRITORIO	22
4.5	EMERGENZE.....	22
4.6	GUASTI E RIDONDANZE.....	24
5.	CONCLUSIONI.....	25
	RIFERIMENTI.....	26

1 INTRODUZIONE

Il presente rapporto tecnico illustra alcuni studi svolti nell'ambito delle attività relative al progetto POR CALABRIA FESR-FSE 2014-2020 denominato Kit-for-Health.

Il campo dell'assistenza sanitaria e della wellness sta costantemente evolvendo e adattandosi alle nuove tendenze tecnologiche. La popolazione mondiale è sempre più costituita da persone anziane che costituiscono una grande porzione dei consumatori dei servizi di assistenza sanitaria e wellness. Per cui, è prevedibile un imminente incremento nella domanda dei servizi di assistenza sanitaria e di wellness. Conseguentemente, si può assumere che ci sarà un trend di crescita nella domanda dei professionisti che si occupano dell'assistenza sanitaria, domiciliare e del wellness nel senso più ampio del termine. La futura tendenza sarà quella di ricevere l'assistenza anziché nelle strutture di assistenza specializzate, molti fruitori di questi servizi preferiranno restare nel loro ambiente familiare.

Nuove tecnologie e metodologie possono essere applicate agli attuali modelli sanitari per ottenere i livelli richiesti dei servizi, data l'infrastruttura a disposizione. Per aumentare l'efficienza del processo di gestione operativo, è naturale utilizzare tecnologie informatiche e delle telecomunicazioni. Le applicazioni vanno da tecnologie di base per eseguire task di gestione giornalieri al supporto comprensivo alle decisioni critiche usando sistemi software specializzati. In particolare, l'uso di metodi del campo dell'intelligenza artificiale (AI), come i sistemi software distribuiti, è un trend crescente nelle applicazioni pratiche, incluso l'uso di agenti software. Un agente è un'entità software che è capace di percepire il suo ambiente e agire autonomamente.

In letteratura esistono diverse definizioni di caratteristiche che fanno riferimento agli agenti, queste possono essere riassunte nelle seguenti proprietà:

- **Autonomia:** gli agenti possono decidere di svolgere compiti o attività anche complessi senza bisogno dell'intervento umano;
- **Abilità sociali:** gli agenti sono in grado di comunicare con i loro simili per coordinarsi e per completare una certa attività;
- **Reattività:** a seconda del contesto in cui opera, un agente è in grado di reagire in modo appropriato e coerente al contesto.

Esistono diverse piattaforme per lo sviluppo di sistemi multi agente, tra queste per la realizzazione del lavoro proposto è stato utilizzato JADE (Java Agent Development Framework), realizzato in Java che supporta lo sviluppo di applicazioni distribuite, fornendo un insieme di servizi di base, conformi allo standard FIPA1.

In questo contesto l'uso dei sistemi multiagente (MAS) e dei simulatori basati sull'uso di agenti è molto promettente. Essi possono essere usati in molti modi diversi, ad esempio, un sistema software distribuito basato su agenti può essere usato per supportare la coordinazione automatica degli operatori socio-assistenziali nelle loro operazioni. In particolare, la tecnologia multiagente è conosciuta per offrire soluzioni flessibili e adattive nei sistemi IT. Inoltre, i sistemi di assistenza che sfruttano componenti simulazioni basate su agenti sono capaci di fornire supporto alle decisioni sulla base

dell'esecuzione delle simulazioni, che tentano di imitare il comportamento dei sistemi reali. Valutare la fattibilità di alcune soluzioni su un modello simulato del mondo reale è sicuramente meno costoso e sicuramente fa risparmiare molto tempo. L'uso multiplo di agenti come un paradigma di modellazione per costruire società artificiali o sistemi sociali è un modo unico per testare teorie per diversi domini applicativi. Un sistema multiagente consiste di un certo numero di agenti che interagiscono tra loro, tipicamente scambiandosi messaggi attraverso una infrastruttura di rete. Nel seguito il termine MAS è usato per descrivere un sistema software distribuito e distinguerlo da un sistema software che fa uso della simulazione basata su agenti.

Questo documento focalizza l'attenzione sulle nuove tecnologie sviluppate per fornire la prossima generazione di servizi sanitari e assistenziali in genere. Nello specifico, viene posta attenzione sull'uso della tecnologia Multiagente (MAS), approccio che fornisce una piattaforma potente ad elevate prestazioni per modellare e risolvere problemi reali. Logicamente questa metodologia può essere estesa efficacemente per modellare il campo healthcare e dell'assistenza sanitaria. A tutti i livelli, il dominio dell'assistenza sanitaria è caratterizzato da sistemi di decision making e gestione delle cure condivisi e distribuiti, richiedendo complesse e numerose comunicazioni tra diversi gruppi di professionisti sanitari e assistenziali (medici, infermieri). Per cui, l'idea di entità autonome, intelligenti, proattive, collaborative, interagenti in un ambiente distribuito rende la tecnologia multi-agente il paradigma ideale per modellare e risolvere problemi nel dominio sanitario e assistenziale. In questo lavoro, saranno presentate le caratteristiche e il modello di un sistema multi-agente basato sulla tecnologia Io-T per applicazioni sanitarie e assistenziali.

In questo contesto particolarmente rilevanti sono tutte quelle applicazioni che hanno come obiettivo la wellness soprattutto dei cittadini più deboli come gli anziani consentendo loro di mantenere uno stile di vita indipendente. L'assunzione di base è che quanto più gli anziani sono capaci di vivere in modo sicuro e confortevole in modo autonomo, tanto minore sarà lo stress che ricadrà sui sistemi sanitari e assistenziali.

Il sistema per l'assistenza sanitaria domestica e per la wellness, presentato in questo documento, rappresenta un esempio di questi sistemi. Generalmente sono composti da un numero variabile di sensori disposti sul corpo del paziente organizzati in un body area network (BAN) solitamente wireless. Lo scopo di questo tipo di sistemi è quello di monitorare le funzionalità del paziente senza impedirne i movimenti. Per la rete di sensori sono possibili due diverse configurazioni: nella prima i sensori inviano i propri dati ad una piccola unità centrale portata dal paziente che li memorizza e successivamente li deposita in un server; nella seconda la BAN è agganciata tramite un collegamento wireless a un dispositivo fisso che memorizza i dati provenienti dai sensori.

Per quanto riguarda la rilevazione delle emergenze sono possibili, in questo scenario, diverse strategie. Le più diffuse in letteratura sono principalmente di due tipi: l'analisi dei dati in place, ovvero il dispositivo che raccoglie i dati dei sensori, li analizza e decide se vi sono le condizioni per lanciare l'emergenza; oppure l'analisi dei dati dopo che questi sono stati trasmessi e memorizzati da un'unità esterna alla BAN come ad esempio un server. Entrambi i metodi offrono vantaggi e svantaggi: nel caso dell'analisi dei dati in place i vantaggi consistono nella velocità di rilevazione, mentre, dato che questa analisi viene limitata ad un ristretto numero di valori per le scarse capacità di memoria del

dispositivo, potrebbero verificarsi dei falsi allarmi; nel caso dell'analisi dei dati in remoto, disponendo di più dati, si avrebbe un minor numero di falsi positivi, tuttavia i tempi di rilevazione delle emergenze si allungerebbero.

In ambito Wellness le soluzioni offerte permettono ad utenti dediti ad attività fisiche di essere sempre aggiornati, in tempo reale, circa il loro stato di salute ed il loro stile di vita andando a definire allenamenti e cicli di attività in modo totalmente personalizzato. L'elevato livello di integrazione della piattaforma rappresenta la principale caratteristica innovativa implementata a livello di servizio: i dati raccolti attraverso il monitoraggio remoto dei parametri fisici (e.g. battito cardiaco, saturazione, pressione sanguigna, peso corporeo, etc.) ed ambientali (e.g. movimento, permanenza sulla sedia, apertura/chiusura porta, etc.) definiscono un profilo comportamentale. L'analisi incrociata dei dati definisce l'impatto che gli scostamenti nelle abitudini possono avere sull'utente. In tal modo si possono prevenire possibili problemi attraverso la pianificazione di azioni mirate e processi di feedback da parte di professionisti di settore (e.g. operatori dei Servizi Sociali, Medici di Medicina Generale, Personal Trainer, etc.). Le applicazioni disponibili affrontano processi e tematiche relative ad ambiti quali l'Invecchiamento Attivo, l'Inclusione Sociale, l'Assistenza Socio-Sanitaria Integrata, il Wellness on the GO, ampliando il bacino d'utenza raggiungibile e la corrispondente rete di servizi proposti, proprio grazie alla valorizzazione dell'IoT. I sistemi proposti in tale ambito ricadono nelle seguenti categorie: telemedicina, daily living e monitoring, rilevazione di emergenze, e assistenza.

Telemedicina.

Le applicazioni in questa categoria sono capaci sia di fornire monitoraggio remoto di pazienti biomedicali sia di comunicare con i sanitari. Sensori distribuiti sul corpo del paziente facilitano il monitoraggio di parametri vitali e biologici. Questi parametri possono essere resi disponibili in tempo reale agli operatori sanitari e assistenziali che sono capaci di comunicare direttamente con il paziente. Le comunicazioni possono avvenire attraverso diversi strumenti come il telefono, via e-mail, o attraverso tecnologie di video conferenze. Tali applicazioni sono state proposte per fornire ai pazienti non indipendenti che si trovano in zone remote accesso ai servizi sanitari.

Daily living e monitoring

Le applicazioni in questa categoria usano sensori wearable e impiantabili per un monitoraggio discreto dei segnali vitali dei pazienti. Tipicamente, questi sistemi sono progettati per mantenere informazioni storiche, fornire approfondimenti e riportare anomalie. Patologie come diabete, tumore, o disabilità fisiche possono beneficiare da questo tipo di applicazioni.

Rilevazioni e assistenza

Le applicazioni in questa categoria sono simili alle applicazioni di daily living e monitoring, con la funzionalità aggiuntiva di fornire assistenza ai pazienti, quando necessario. Per esempio, un'applicazione per rilevare e identificare cadute e fornire assistenza potrebbe aiutare persone anziane con difficoltà a camminare ad avere una vita indipendente.

Il fine ultimo di queste applicazioni con tecnologia MSAS è prolungare lo stile di vita indipendente e autonomo degli anziani, riducendo il costo complessivo del sistema sanitario.

Nel documento verranno descritti la gestione di reti di sensori, gli applicativi dal lato medico e, in generale, del personale addetto all'erogazione delle prestazioni wellness nonché, per gestire le emergenze, anche quelli relativi al personale medico di supporto. Infine il documento propone una piattaforma ad agenti basate su tecniche IoT.

Per quanto riguarda i sistemi assistenziali e sanitari, lo studio si è focalizzato su due tecnologie: il cloud computing e i sistemi multi-agente (MAS). In [1] viene proposto un sistema mobile che permette la memorizzazione, l'aggiornamento e l'accesso a dati elettronici relativi all'assistenza sanitaria utilizzando il cloud computing. In [2] viene utilizzata una rete di sensori wireless per automatizzare il processo di raccolta dei dati. Le informazioni sono distribuite al personale medico grazie a tecnologie cloud. Koufi e al. [3] propongono un approccio alternativo, dove l'interoperabilità di dati e servizi è ottenuta mediante un sistema multi-agente.

Il resto del documento è strutturato come segue. La sezione 2 fornisce una rassegna dei sistemi multi-agente. La sezione 3 presenta lo stato dell'arte dei sistemi IoT per applicazioni health e wellness. La sezione 4 illustra uno scenario applicativo di un sistema multi-agente pervasivo di risposta a un'emergenza sanitaria con assistenza domestica. Infine, nella sezione 5 sono discussi i risultati di questo lavoro.

2. TECNOLOGIE ABILITANTI AGENT-BASED

La stretta integrazione tra mondo digitale e fisico è resa possibile dai recenti progressi nella microelettronica, nei sistemi di telecomunicazione, nei sistemi embedded, nei protocolli di comunicazione e negli algoritmi per l'analisi dei dati, come di seguito dettagliato.

2.1 MINIATURIZZAZIONE DELL'HARDWARE ED EFFICIENZA ENERGETICA

In passato, il concetto di computer era associato a macchine stazionarie e collegate ad alimentatori ad alta potenza, e lo sforzo di progettazione si concentrava nell'ottenere performance sempre più elevate in termini di frequenza di lavoro e potenza di calcolo. Dagli inizi del 2000, con la fine della validità della legge di Dennard [4], e l'aumento nella domanda per dispositivi mobili, l'industria si è concentrata sulla progettazione di dispositivi a basso consumo ed energeticamente più efficienti, seppur performanti [5]. Oggi tutti i principali produttori rendono disponibili dispositivi e *system-on-a-chip* che possono essere facilmente integrati in oggetti e cose, grazie alle loro piccole dimensioni ed ai loro consumi ridotti. La potenza computazionale di questi dispositivi è, ovviamente, non comparabile con quella delle unità di elaborazione delle workstation o dei server, ma è sufficiente per fornire capacità computazionale general purpose agli oggetti che li integrano.

2.2 TECNOLOGIE DI RETE

I sistemi cyber-physical e l'Internet delle Cose sono basati su componenti connesse, che possano essere dispiegate su aree piccole, come una singola stanza od una casa, o aree più grandi, come reti elettriche od altre infrastrutture urbane. Le grandi industrie di telecomunicazioni e le comunità di ricerca hanno sviluppato tecnologie che consentono una comunicazione affidabile, utilizzando mezzi wired o wireless, che possano essere sfruttati su aree di dimensione diversa. La connettività Internet

wireless è ormai garantita quasi universalmente dalle reti mobili digitali, con standard tecnologici che si sono evoluti negli anni dal GSM, fornendo sempre più capacità di banda [6]. Lo standard correntemente in sviluppo, LTE-advanced, promette picchi fino a 100Mb/s per gli utenti [7]. Per quanto riguarda le reti locali, lo standard corrente IEEE 802.11, denominata 802.11ac, presenta un avanzamento sia in termini di velocità di trasmissione che in efficienza energetica [8]. È inoltre in stato di draft lo standard 802.11ah, che mira a ottenere coperture fino a 1 km alla velocità minima di 100kbps e consumi energetici inferiori, cose che ne consentirebbe l'utilizzo per reti wireless di sensori e attuatori e per l'Internet delle Cose anche su scala metropolitana. Nelle reti personali, il riferimento è ad ora rappresentato dalle specifiche ZigBee e Bluetooth. Il primo è uno standard per reti mesh a basso consumo e a basso costo, che mira a consentire l'impiego di un gran numero di dispositivi alimentati a batteria e a lunga durata per applicazioni di monitoraggio e controllo [9]. Esso è, per esempio, largamente usato per applicazioni di domotica e Smart Home. Bluetooth fornisce un data rate maggiore, se comparato a quello fornito da ZigBee, al costo di una minore distanza di trasmissione ed un maggiore consumo energetico [10]. La tecnologia Bluetooth è attualmente usata per stabilire comunicazioni single-hop tra due dispositivi per scambiare flussi o dati multimediali.

Visto il grande numero di dispositivi che possono essere coinvolti in un sistema cyber-physical integrato, problematiche di rete rilevanti che necessitano di essere citate sono quelle dell'indirizzamento e dell'identificazione univoca di dispositivi e risorse.

Nel 1999 l'espressione Internet of Things fu introdotta da Ashton [11] in una presentazione sulla tecnologia Radio Frequency Identification (RFID) [12]. La tecnologia RFID consente, tra le altre cose, il riconoscimento immediato, da parte di computer od altri dispositivi, di oggetti passivi opportunamente etichettati con appositi tag. La specifica Electronic Product Code Tag Data Standard (EPC-TDS) [13] descrive, come e quali informazioni codificare in un tag RFID per identificare un certo oggetto univocamente, descrivendone al contempo alcune caratteristiche e proprietà. Estensioni a questa specifica sono state proposte nel tempo, ad esempio in [14], per poter meglio affrontare le problematiche di apertura ed interoperabilità legate al contesto dell'Internet delle Cose.

Problematiche legate all'indirizzamento sono state risolte, per esempio, attraverso l'introduzione dell'Internet Protocol Version 6 (IPv6), che sfrutta uno spazio degli indirizzi estremamente più grande rispetto a quello del suo predecessore. Un indirizzo IPv6 è costituito da 128 bit, dal quale si possono ottenere fino a circa 3.4×10^{38} indirizzi, che si stima siano più che sufficienti per soddisfare il bisogno di indirizzi unici per i prossimi decenni. È inoltre interessante notare che specifiche quali ZigBee e Bluetooth forniscono estensioni per integrare i loro stack con il protocollo IPv6 [15].

2.3 COMPUTER SCIENCE

Tecnologie abilitanti dal punto di vista informatico coprono un'ampia gamma di aree di ricerca. I primi calcolatori general purpose erano progettati allo scopo di eseguire calcoli e operazioni matematiche, e l'unico loro requisito era la correttezza delle elaborazioni e dei risultati ottenuti. Quando i computer iniziarono ad essere usati per controllare sistemi fisici, la sola correttezza risultò non essere sufficiente, poiché, in questi contesti, è necessario tenere conto di problematiche legate alla gestione del tempo. Questo incoraggiò la progettazione e lo sviluppo di macchine, modelli ed

algoritmi che consentissero l'elaborazione in tempo-reale [16], che cioè rispondessero al problema di come schedare task computazionali in modo che ogni compito in ogni task fosse completato rispettando la sua deadline. Molte applicazioni cyber-physical prevedono automazione e attuazione su ambienti distribuiti, in soft o hard real time.

I sistemi cyber-physical e Internet of Things possono inoltre essere visti come naturale evoluzione delle Wireless Sensor Networks (WSN). Il concetto di *mote*, come piccolo nodo dotato di capacità di sensing, computazione e comunicazione, fu introdotto nel progetto Smart Dust [17]. I mote possono integrare un insieme di sensori, e sono in grado di trasportare informazioni sull'ambiente nel quale sono immersi attraverso la rete formata dall'insieme dei mote tra loro interconnessi. Quando questi nodi hanno iniziato ad ospitare non solo sensori, ma anche attuatori si è passati al concetto di Wireless Sensors and Actuator Network (WSAN) [18], che può essere vista come una rete di controllo distribuita molto vicina ad un sistema cyber-physical.

Ogni dispositivo immerso in uno Smart Environment può produrre dati dal suo stato interno e da misure sull'ambiente dove esso è situato. Tali dati possono essere prodotti a velocità anche elevata, conseguentemente, sono necessarie tecniche per gestire le grandi quantità di dati ed i flussi prodotti da sistemi anche di larga scala. Data la mole di informazioni, soluzioni che richiedono (i) che i dati siano conservati da qualche parte e disponibili nella loro interezza e (ii) programmi ed algoritmi complessi in grado di effettuare elaborazioni esatte possono non essere applicabile. Quando la dimensione dei dati è molto grande, come nei sistemi cyber-physical, sono richieste tecniche e algoritmi che sacrificino ragionevolmente la precisione dei risultati per soddisfare i requisiti di elaborazione, sia in termini di memoria utilizzata che di tempo impiegato. Modelli e algoritmi per elaborare su flussi di dati (*data stream*) sono esaminati, per citarne solo alcuni lavori, in [19], dove il concetto di data stream è definito formalmente e sono presentati algoritmi efficienti per analisi ed aggregazioni statistiche.

Un algoritmo su data stream è, generalmente, capace di eseguire elaborazioni online, producendo o aggiornando il suo output non appena un nuovo elemento è disponibile dal flusso. Questo richiede che l'algoritmo processi ogni elemento una volta (ed un numero definito di volte), e poi scarti e rimuova dalla memoria quell'elemento. In alcune applicazioni, però, può sussistere la necessità di conservare i dati per ulteriori analisi offline e usi futuri. La ricerca nel campo dei "Big Data" [20] si occupa di indagare l'insieme delle tecniche per affrontare le problematiche della conservazione, processamento ed estrazione di conoscenza di quantità anche enormi di dati, e tutte queste tecniche sono da tenere in considerazione nel progettare e implementare sistemi intelligenti [21].

2.4 PIATTAFORME SMART O CYBERPHYSICAL

Gli Smart Object (SO), importanti component della IoT, sono oggetti di uso comune o quotidiano equipaggiati con componenti hardware quali CPU, moduli di comunicazione e sensori/attuatori. L'obiettivo è di far diventare questi oggetti *coscienti* dell'ambiente in cui sono situati e di permetterne una sorta di *controllo* sulle basi di specifici obiettivi e capacità dell'oggetto stesso.

In generale, lo sviluppo di SO è legato a diverse problematiche quali ad esempio la comunicazione fra SO, l'interfacciamento con sensori e attuatori, proattività, knowledge management e

computazione distribuita. Al fine di facilitare il processo di sviluppo, diverse attività di ricerca sono rivolte alla definizione di nuovi framework e middleware che favoriscano una prototipazione rapida di SO. Questi middleware normalmente forniscono delle API ben definite tramite le quali nuovi SO possono essere programmati e messi in esercizio.

Gli SO sono in grado di collaborare e di supportare quella che viene definita come Ambient Intelligence al fine di realizzare *Smart Environment* (SE) [22]. Uno SE può essere definito come un ambiente cyber-fisico aumentato con una collezione di sistemi embedded capaci di elaborare dati fra di loro eterogenei e di interagire con le persone. Anche molteplici middleware per una rapida prototipazione di SE sono stati proposti in letteratura e attualmente disponibili [23]. Di seguito è fornita una panoramica dei principali middleware usati per lo sviluppo di applicazioni basati su SO e per gli SE.

2.5 SISTEMI MULTI-AGENTE

La programmazione basata ad agenti rappresenta un promettente approccio per lo sviluppo di applicazioni complesse in ambiente distribuito in cui è necessario operare senza una forma di controllo centralizzato. Tale paradigma di programmazione fornisce un modo per progettare e sviluppare applicazioni in termini di entità autonome, riferite come *agenti*, che sono situati in un ambiente e che perseguono i loro obiettivi in modo flessibile interagendo fra di loro per mezzo di protocolli e linguaggi di alto livello [24].

Se confrontato con altri paradigmi di programmazione, quello ad agenti può essere definito come una evoluzione di altri già esistenti. Da un punto di vista di utilizzo pratico, esso può essere invece considerato come una vera e propria rivoluzione. Attualmente la programmazione ad agenti viene utilizzata in molteplici campi applicativi [25] anche molto eterogenei. Il loro uso spazia da piccole applicazioni, quali ad esempio filtri per la posta elettronica, alla realizzazione di sistemi di controllo complessi e tempo-dipendenti quali ad esempio i sistemi di controllo del traffico aereo. Knowledge Management, Team Working, Bio-Inspired Architecture, Supply Chain Management, Entertainment and Health Care sono campi applicativi in cui la ricerca e l'utilizzo di sistemi basati su agenti è attualmente di notevole interesse. Più in particolare, gli agenti rivestono un ruolo chiave nello sviluppo di applicazioni in cui il comportamento del sistema da realizzare è determinato principalmente dalle interazioni (complesse) tra le entità costituenti il sistema stesso e dove diventa estremamente difficile, o addirittura impossibile, descrivere il comportamento del sistema nella sua interezza.

Un agente è una astrazione software di alto livello particolarmente indicata per lo sviluppo di applicazioni software moderne in quanto: (i) la sua capacità di operare autonomamente riflette intrinsecamente la natura decentralizzata dei sistemi distribuiti, (ii) costituisce la naturale evoluzione di strumenti di astrazione quali modularità e incapsulazione, (iii) il modo flessibile con cui gli agenti interagiscono (sia tra di loro e sia con l'ambiente che li ospita) permette loro di operare in scenari dinamici e contesti non completamente prevedibili a priori.

Il termine agente significa letteralmente *colui che agisce* o *colui che fa qualcosa*. Una tacita assunzione è quella per cui un agente *riveste ruoli attivi e promuove azioni* che possono influenzare altri agenti e l'ambiente circostante. Una definizione classica del termine agente è quella data da Wooldrige and Jennings [26]: "... il termine agente è usato per denotare un componente hardware o, più frequentemente, un componente software che esibisce le seguenti proprietà:

- *autonomia*: gli agenti operano senza il diretto intervento di esseri umani e possiedono controllo sulle loro azioni e sul loro stato interno;
- *abilità sociali*: gli agenti interagiscono con altri agenti (e possibilmente con le persone) tramite l'uso di appropriati linguaggi di comunicazioni (agent-communication language);
- *reattività*: gli agenti percepiscono l'ambiente loro circostante (l'ambiente può essere sia quello fisico che virtuale) e reagiscono ai cambiamenti che in esso si verificano."

Le proprietà fino ad ora riportate forniscono quella che viene definita come *nozione debole* di agente. In alcuni contesti, tuttavia, una definizione più forte di agente è necessaria. Una *nozione forte* di agente può essere ottenuta tenendo conto di alcune (o di tutte) le proprietà di seguito elencate:

- *mobilità*: riferisce all'abilità di un agente di muoversi all'interno di una rete di calcolo;
- *veracità*: è l'assunzione per cui un agente non fornisce mai informazioni ritenute essere false;
- *benevolenza*: è l'assunzione per cui un agente non ha obiettivi contrastanti rispetto a quelli che gli sono stati chiesti di perseguire;
- *razionalità*: è l'assunzione per cui un agente opera sempre in modo da perseguire i suoi obiettivi e che non agirà in modo da ostacolarli. Tutto ciò coerentemente alle conoscenze che l'agente possiede circa sé stesso e l'ambiente che lo circonda.

Un agente può essere considerato come un problem-solver che opera in un contesto ben delimitato e che possiede/offre un insieme ben definito di funzionalità. Oltre che per sé stesso, esso può operare al posto di, o per conto di, un altro agente o di un utente finale. Gli agenti sono situati (o meglio immersi) in un ambiente sul quale possiedono un controllo e una visibilità parziale. Un agente percepisce l'ambiente tramite i suoi *sensori* e agisce su di esso tramite i suoi *attuatori*. Un ambiente definisce le proprietà del mondo (ad esempio, leggi, regole, vincoli e politiche) per cui un agente può operare. Un sistema in cui più agenti sono coinvolti è solitamente riferito come Multi Agent System (MAS). Un MAS può essere visto come una rete debolmente connessa di problem-solver (ovvero di agenti) che interagiscono per affrontare problematiche che andrebbero oltre le capacità e le conoscenze individuali dei singoli individui che, dal loro canto, possiedono solo una limitata visione delle problematiche stesse. In questo modo, non esiste una sorta di controllo centralizzato e le computazioni sono intrinsecamente asincrone. La nozione di ambiente (esplicitamente considerata), insieme alle capacità sociali, sono le caratteristiche fondamentali che distinguono la programmazione ad agenti rispetto ad altri paradigmi quali ad esempio quelli ad oggetti [27].

Rispetto ad altri paradigmi, la metafora degli agenti fornisce un modo efficace ed efficiente per descrivere e modellare propriamente scenari appartenenti al mondo reale che ci circonda e di

affrontare situazioni in cui il comportamento collettivo e le interazioni fra individui produce un comportamento emergente che non può essere attenuato dai soli singoli individui [28].

Ogni MAS necessita di una infrastruttura a runtime, o piattaforma di esecuzione, per l'esecuzione dei sistemi realizzati. Una piattaforma ad agenti è uno strato di middleware che fornisce un *execution locus* indipendente dalla reale piattaforma hardware/software (ovvero nodo computazionale) usata per l'esecuzione del software. Una piattaforma di esecuzione per gli agenti fornisce supporto per l'esecuzione vera e propria degli agenti, per gestire la loro esecuzione, per accedere alle risorse di sistema, per garantire integrità e protezione agli agenti e alla piattaforma stessa. Una piattaforma fornisce servizi per la comunicazione, la migrazione e per il naming degli agenti. Una piattaforma può inoltre favorire l'uso di specifiche metodologie ed offrire strumenti per lo sviluppo di applicazioni.

Un grande numero di piattaforme ad agenti sono attualmente esistenti. Esse provengono sia da contesti accademici che industriali e molte di essi sono state sviluppate utilizzando tecnologia Java. Un tentativo per la standardizzazione delle piattaforme ad agenti ha portato alla definizione di due principali standard che sono MASIF [29] e FIPA [30]. Gli agenti che appartengono a specifiche piattaforme che afferiscono a questi standard possono collaborare al fine di perseguire scopi e obiettivi comuni tramite lo scambio di informazioni (messaggi). Tuttavia gli agenti possono solo migrare ed essere eseguiti in piattaforme compatibili e ritenute potenzialmente sicure. Inoltre una piattaforma accetta la migrazione su di essa di un agente solo se l'agente stesso è considerato come affidabile. Tutto ciò implica che la visione per cui un agente sia in grado di muoversi liberamente su piattaforme disponibili in rete non è ancora realistica. Di seguito vengono descritte alcune piattaforme ad agenti.

JADE

JADE¹, sviluppata dai laboratori TLAB, è una piattaforma open Sorgente per lo sviluppo di applicazioni distribuite che sfruttano agenti peer-to-peer. Essa è completamente implementata in Java. JADE semplifica lo sviluppo di applicazioni ad agenti fornendo uno strato middleware che è conforme allo standard FIPA [30] e fornendo un insieme di strumenti grafici che supportano le fasi di debug e deployment delle applicazioni. La configurazione delle applicazioni è ottenuta tramite una GUI remota.

JACK

JACK² è un esempio di piattaforma ad agenti di uso commerciale. Essa incorpora una suite di strumenti grafici finalizzati allo sviluppo e all'analisi delle applicazioni realizzate nonché alla realizzazione, esecuzione ed integrazione di applicazioni basate su agenti e che sfruttano un approccio di sviluppo di tipo modulare. Lo sviluppo degli agenti si basa sul linguaggio ad agenti JACK che estende Java con concetti di tipo agent-oriented.

¹ <https://jade.tilab.com/>

² <http://www.agent-software.com.au/products/jack/>

James

JAMES³ è una piattaforma sviluppata interamente in Java finalizzata soprattutto allo sviluppo di applicazioni relative al campo della modellazione e simulazione. Essa è basata su una architettura in cui moduli esterni possono essere facilmente aggiunti. Tutto ciò conferisce estrema flessibilità nel realizzare applicazioni. L'architettura modulare permette l'uso di molteplici tecniche di modellazione e simulazione che possono così essere facilmente integrate e sfruttate all'interno del framework offerto. In aggiunta, JAMES mette direttamente a disposizione l'uso di algoritmi e strumenti finalizzati alla realizzazione di applicazioni efficienti. In aggiunta, la piattaforma offre pieno controllo sugli esperimenti simulativi da effettuare e sui parametri da fornire in input ad ogni simulazione in modo da replicarne l'esecuzione in diverse condizioni. JAMES è stato utilizzato in molteplici aree applicative e su architetture hardware che offrono differenti potenze di calcolo (dal semplice notebook ai performanti cluster di workstation).

Nome	Sviluppato da	Dominio applicativo di riferimento	Licenza	Open Sorgente
Agent Factory	University College Dublin	General purpose agent based	LGPL	Yes
AgentBuilder	Acronymics Inc.	General purpose multi-agent systems	Proprietary, Discounted academic licenses	No
AgentScope	Delft University of Technology	Large-scale distributed agent systems	BSD	Yes
AGLOBE	Czech Technical University	Real-world simulations	LGPL	Yes
AnyLogic	The AnyLogic Company	General purpose, distributed agent based simulations	Commercial, Academic license	No
Cormas	Cirad research centre	Natural reSorgentes and agent-based simulations	GPL	Yes
Cougaar	Raytheon BBN Technologies	Complex, large-scale, distributed applications	Cougaar Open License Sorgente License COSL	Yes
CybelePro	Intelligent Automation Inc	Large-scale distributed systems	Commercial, Academic license	No
EMERALD	LPIS Group, Aristotle University of Thessaloniki	Distributed applications composed of autonomous entities	LGPL	Yes
GAMA	IRD/UPMC International Research Unit UMMISCO	Large-scale distributed spatially explicit agent-based simulations	GPL	Yes
INGENIAS Development Kit	grasia! research group, Universidad Complutense Madrid	General purpose agent based	CC By-SA GPLv2	Yes
JACK	AOS	Dynamic and complex environments	Commercial, Academic license	No
JADE	Telecom Italia (TILAB)	Distributed applications composed of autonomous entities	LGPLv2	Yes
Jadex	Hamburg University	Distributed applications composed of autonomous BDI entities	LGPLv2	Yes

³ <https://james.apache.org/>

JAMES II	University of Rostock	General purpose agent based modeling and simulations	JAMESLIC (compatible with GPL)	Yes
JAS	Università di Torino	General purpose agent based	LGPL; 3 rd party licenses	Yes
Jason	Universities of Rio Grande do Sul and Santa Catarina	Distributed applications composed of autonomous BDI entities	LGPLv2	Yes
JIAC	Technische Universität Berlin	Large-scale distributed systems	Apache License V2	Yes
MaDKit	Institut universitaire de technologie	Multi-agent systems with agent based simulation	GPL	Yes
MASON	George Mason University	Event-driven multiagent simulations	Academic Free License version 3.0	Yes
<i>NetLogo</i>	The Center for Connected Learning (CCL) and Computer-Based Modeling, Northwestern University	Agent-based simulations	GPL	No
Repast	University of Chicago	Agent-based simulations	New BSD	Yes
SeSAM	Örebro University	Agent-based simulations	LGPL	Yes
Swarm	Swarm Development Group	General purpose agent based	GPL	Yes

Tab. 1– Piattaforme per lo sviluppo di applicazioni basate ad agenti

3 SISTEMI IOT PER APPLICAZIONI HEALTH E WELLNESS

La tecnologia IoT è sempre più utilizzata nell'ambito delle applicazioni wellness e dell'healthcare in generale. I progetti sviluppati riguardano diversi ambiti: si va ad esempio da quello strettamente domestico a quello ospedaliero fino a quello per l'assistenza agli anziani.

3.1 ARCHITETTURA DI BASE

L'architettura generale di un sistema IoT per tale tipo di applicazioni si compone di una serie di sensori collegati a uno o più dispositivi anche dislocati geograficamente. Il collegamento è solitamente wireless e si utilizzano reti con bassa copertura come ad esempio Bluetooth o ZigBee.

I sensori che vengono generalmente più utilizzati in questi sistemi sono i seguenti:

- accelerometro a doppio asse: sensore che misura le variazioni di acceleratore a cui è sottoposto il soggetto in studio. La misurazione viene effettuata rilevando l'inerzia di una massa sospesa in un ambiente elastico e sottoposta ad accelerazione. Con l'utilizzo di questo sensore è possibile stabilire l'attività del paziente, come verificare una caduta quando il valore misurato è molto elevato
- termometro: un sensore in grado di rilevare la temperatura corporea o le sue variazioni quando questo è posto a contatto con la cute del paziente.
- cardiografometro: sensore in grado di rilevare la frequenza cardiaca in tempo reale, utile per verificare eventuali situazioni di stress cardiache.
- ECG (elettrocardiogramma): sensore che misura l'attività elettrica del cuore, di solito la misurazione avviene tramite una serie di elettrodi che vengono posti sul torace del paziente. Il segnale trasmesso genera un tipico tracciato composto dall'onda P, dal complesso QRS, dall'onda T ed eventualmente dall'onda U. Dallo studio di questo tracciato possono essere evidenziate aritmie cardiache.

- Sensore SpO2: misura la pressione di saturazione dell'ossigeno a livello ematico tramite un sistema di rilevamento a luce rossa o infrarossa. È solitamente utilizzato per rilevare problemi cardiorespiratori.
- sfigmomanometro: è un sensore in grado di rilevare la pressione minima (diastolica) e massima (sistolica). Generalmente è composto da un manicotto collegato ad un dispositivo elettronico legato al braccio del paziente e, per funzionare, necessita del cardiofrequenzimetro, poiché la misurazione è basata sulla variazione del battito cardiaco quando il manicotto viene gonfiato e sgonfiato. La rilevazione richiede generalmente fra i 10 e i 20 secondi quindi, a differenza dei sensori precedenti, non può essere effettuata in real time.

Oltre ai sensori precedentemente descritti e disposti sul corpo dei pazienti, molto diffuso è l'uso di sensori ambientali che monitorano l'ambiente intorno alle persone (come i pazienti) sotto esame.

La rete dei sensori può essere collegata ad uno o più server che memorizzano i dati e consentono analitica avanzata degli stessi.

La maggior parte delle applicazioni presenti in commercio presentano caratteristiche analoghe come ad esempio la possibilità di visualizzare i valori dei parametri dei sensori per quanto riguarda il paziente e offrono capacità di analisi mediche e da parte di operatori avanzate dal punto di vista medico.

a. AMON

Il primo sistema presentato nel dettaglio è Amon, un sistema di monitoraggio e allarme medico basato su più parametri che utilizza un dispositivo indossabile, progettato soprattutto per pazienti con gravi problemi cardiorespiratori.



FIGURA: PROTOTIPO AMON

Dal punto di vista hardware, Amon si compone di due dispositivi: il primo a forma di bracciale che incorpora diversi sensori ed è in grado tra l'altro di misurare la pressione arteriosa, l'ECG e l'attività del paziente tramite un accelerometro a doppio asse. Inoltre fornisce un piccolo led che permette di visualizzare le varie misurazioni effettuate. Il secondo dispositivo è una unità stazionaria situata presso il centro di telemedicina (TMC). I due dispositivi sono connessi mediante un collegamento cellulare

GSM/UMTS e la comunicazione avviene tramite l'invio di SMS. I dati raccolti vengono consegnati al TMC tre volte al giorno.

La rilevazione delle emergenze può essere svolta sia in place sia in modo remoto. In place vengono semplicemente stabilite delle soglie per ogni parametro preso in considerazione: quando uno di questi valori supera tale soglia scatta l'emergenza. In modo remoto invece vengono svolti calcoli specifici per ogni parametro che si vuole prendere in considerazione in modo da gestire eventuali emergenze non rilevate dal dispositivo.

Uno degli aspetti positivi del sistema è senz'altro l'utilizzo di un unico dispositivo che, in modo non intrusivo, è in grado di rilevare eventuali anomalie tempestivamente e in modo aggressivo.

b. WAITER

WAITER, come nel caso di Amon, è un sistema che consente di rilevare eventuali anomalie. Le misurazioni dei parametri vitali del paziente avvengono mediante un apparecchio di piccole dimensioni che viene posto dietro il padiglione auricolare ed è costituito da tre sensori, il primo misura la frequenza cardiaca, il secondo la temperatura e il terzo è un accelerometro a doppio asse per misurare l'attività del paziente. Questo dispositivo è inoltre in grado di connettersi tramite bluetooth per l'invio dei dati raccolti.

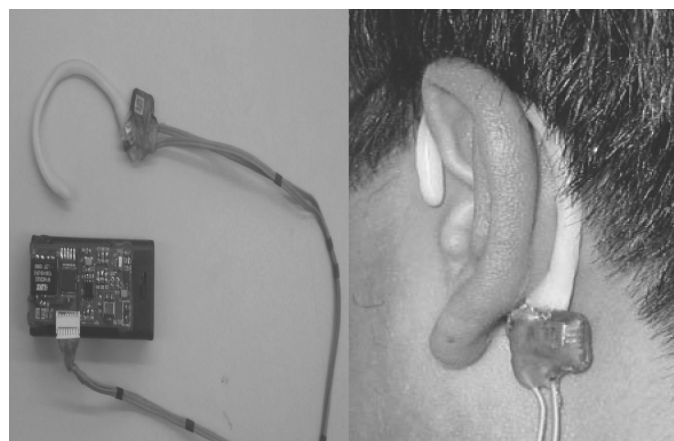


FIGURA: PROTOTIPO WAITER

La novità di questo sistema è che è possibile utilizzare come dispositivo di raccolta e analisi dei dati anche un comune cellulare, dotato di ricetrasmittitore bluetooth, su cui sia installato J2ME (Java 2 Micro Edition).

La raccolta dei dati avviene con frequenza oraria ai databases dislocati negli Healthcare Center (tramite connessione GSM). In questi centri il personal addetto si occuperà di controllare i parametri

dei pazienti. Come in Amon, la rilevazione avviene tramite valori di soglia. Una volta che l'emergenza è lanciata, viene contattato il server dell'Healthcare Center dove il personale addetto provvederà a validarla o meno.

c. SISTEMA AD AGENTI DI K. MILLER E S. SANKARANARAYANAN

In letteratura esistono alcuni sistemi che utilizzano il paradigma ad agenti per gestire un sistema di monitoraggio per l'assistenza sanitaria. Tra questi sistemi uno dei più noti è quello di K. Miller e S. Sankaranarayanan, progettato per essere utilizzato in ambito ospedaliero, anche se molte sue soluzioni potrebbero essere applicate anche in ambito wellness più in generale e in quello domestico.

La misurazione dei parametri vitali avviene tramite sensori che inviano i loro dati ad un'unità non necessariamente propria del paziente (per esempio, possono essere inviati su un dispositivo mobile appartenente al personale infermieristico dell'ospedale) che poi li inoltra al database dove vengono memorizzati e gestiti. La rilevazione delle anomalie o emergenze, avviene sia in place, sia in remoto, stabilendo dei valori di soglia per ogni misurazione dei sensori.

3.2 ANALISI DEI REQUISITI DEL SISTEMA

Il problema preso in considerazione è quello di fornire un servizio di monitoraggio e di assistenza medica per problemi fisici non gravi in modo da poter evitare, per quanto possibile, il ricovero ospedaliero utilizzando apparecchi che non limitino la libertà di movimento della persona che li utilizza.

Gli utilizzatori del sistema saranno essenzialmente i pazienti e il personale esperto come medici e gli operatori sanitari sul territorio (d'ora in poi chiamati per brevità OST).

- ***Richieste del paziente***

Generalmente i dispositivi di monitoraggio non devono intralciare le normali attività giornaliere di una persona: devono essere piccoli e semplici da utilizzare e, con pochi comandi, si vuole mostrare che i dispositivi funzionino adeguatamente visualizzando in forma semplificata i dati correnti raccolti.

La funzione principale del dispositivo deve essere quella di avvisare tempestivamente, in caso di emergenza, dando al paziente la sicurezza dell'arrivo di soccorsi in tempi rapidi.

- ***Richieste del personale medico***

Generalmente l'utilizzo del sistema da parte del medico avverrà tramite un calcolatore fisso o portatile.

Le applicazioni dal lato medico consentono non solo di visualizzare i dati raccolti periodicamente sui pazienti, ma anche di analizzarli statisticamente e poter seguire la storia clinica accedendo alla cartella clinica.

Oltre a questo, un medico deve essere immediatamente informato se su un paziente viene rilevata un'emergenza. Anche nel caso del medico, infine, potrebbe esserci la necessità di segnalare eventuali allarmi dovuti a situazioni rilevate dopo la consultazione storica dei dati del paziente.

- ***Richieste degli Operatori Sanitari Sul Territorio (OST)***

Per quanto riguarda gli operatori sanitari sul territorio, la loro funzione principale è quella di intervenire in caso di emergenza. Solitamente sui mezzi di intervento è presente del personale qualificato in grado di risolvere eventuali anomalie.

Pertanto l'utilizzo del sistema che viene svolto da parte di un OST è molto simile a quello di un medico: oltre a sapere l'ubicazione del paziente sarà necessario accedere ai dati dei sensori e alla cartella clinica, in particolare ai dati relativi ad eventuali allergie o patologie sofferte dal paziente in modo da evitare complicazioni in un'eventuale somministrazione di farmaci. In genere al termine di ogni soccorso verrà compilato un apposito modulo (ad esempio il modulo 118) contenente le informazioni sull'intervento.

3.2.1 Strumentazione

Per il monitoraggio di un paziente saranno necessari una serie di sensori per acquisirne i vari parametri vitali. Questi sensori necessiteranno, ad esempio, di un trasmettitore bluetooth per inviare i propri dati ad un dispositivo che li analizzi sul posto per poi spedirli ad un server dove verranno memorizzati stabilmente. I sensori previsti sono i seguenti:

- Accelerometro a doppio asse per la misurazione dell'attività del paziente, in grado di rilevare eventuali cadute dello stesso.
- Termometro per la misurazione della temperatura
- Sfigmomanometro per la misurazione della pressione
- Cardiosfigmomanometro per la misurazione delle pulsazioni cardiache.

Per la ricezione dei dati dei dispositivi sarà necessario utilizzare un dispositivo portatile: un esempio potrebbe essere un dispositivo già presente in commercio come nel caso di WAITER, quale un telefono cellulare o un PDA dotato di connessione wi-fi per il collegamento con il database, bluetooth per quanto riguarda la comunicazione con i vari sensori.

Inoltre sarà necessario disporre di uno o più server per il deposito dei dati degli utenti, nonché delle loro cartelle cliniche. Oltre a questo il server potrebbe servire anche per depositare i dati relativi ai medici e al personale OST.

Per quanto riguarda il personale addetto alla sorveglianza come e OST e medici, data la loro necessità di spostarsi, dovrebbero disporre di un dispositivo portatile quale ad esempio un notebook o anche un pda dotato di connessione wireless.

3.3 FUNZIONALITÀ DEL SISTEMA

Le funzionalità del sistema saranno specifiche a seconda della diversa figura individuata: paziente, medico o personale OST.

Lato paziente

L'applicativo paziente, eseguito su un dispositivo mobile, dovrà essere leggero, e l'interfaccia deve prevedere una semplice visualizzazione dei valori dei vari sensori.

Per collegare i vari dispositivi sarà necessario instaurare una piccola rete (ad esempio bluetooth) centralizzata sul dispositivo di raccolta.

Oltre a questo, l'applicativo sul dispositivo dovrà essere in grado di collegarsi al database, ciò richiede anche di instaurare una connessione tcp/ip con il server, nonché disporre delle procedure di accesso in scrittura di dati in un database.

Infine per garantire la privacy dei pazienti, affinché i loro dati sensibili non vengano letti da terze parti non appartenenti al sistema si dovranno stabilire opportuni protocolli di sicurezza come accessi con password, protocolli di criptazione delle comunicazioni fra i dispositivi, ecc.

- ***Lato medico***

L'applicativo al lato medico richiede una gestione più complessa rispetto a quella sul lato paziente. L'interfaccia offerta dovrà essere completa di tutti i dati relativi al paziente, il che comprende informazioni sulle generalità, sulla cartella clinica e sui dati storici e statistici dei sensori. Dovrà quindi fornire accesso in lettura al database per poter scaricare i dati dei sensori e le generalità dei pazienti. Sia in lettura sia in scrittura per poter scaricare e modificare la cartella clinica.

Anche sul lato medico la sicurezza è importante: per evitare che personale non autorizzato acceda a dati protetti dei pazienti sarà necessario gestire opportunamente gli accessi.

Infine per quanto riguarda le emergenze, l'applicativo dovrà poter ricevere, eventualmente per validare o annullare, le segnalazioni di situazioni critiche da parte del paziente e dovrà dare la possibilità al medico di avvisare di eventuali emergenze emerse dall'analisi dei dati presentati.

- ***Lato OST***

Per quanto riguarda l'applicativo utilizzato dal personale OST, esso dovrà essere molto simile a quello medico, offrendo un'interfaccia in grado di mostrare le generalità del paziente, la cartella clinica e i dati dei sensori. Inoltre dovrà essere anch'esso predisposto per accedere sia in lettura sia in scrittura al database: in lettura, come nel lato medico, per ricevere i dati dei sensori, in scrittura/lettura per la cartella clinica ed eventualmente per i moduli post intervento.

4. CASE STUDY: UN SISTEMA MULTIAGENTE PERVASIVO PER L'ASSISTENZA SANITARIA DOMESTICA

In questo capitolo sarà descritto un sistema multi-agente pervasivo per l'assistenza sanitaria domestica presentato e sviluppato in un lavoro di tesi presso l'Università degli studi di Padova.

4.1 SIMULAZIONE DI SENSORI

Non disponendo di reali sensori per la realizzazione del progetto in tesi, l'autore ha utilizzato delle interfacce realizzate in Java che consentissero l'interazione per la modifica dei parametri presi in esame.

Per realizzare l'accelerometro si è utilizzato Java3D.

Dato che un accelerometro misura l'accelerazione a cui è soggetto un corpo, questo è stato rappresentato da una sfera in uno spazio 3D. Nella realizzazione della sfera si è usata una primitiva di Java 3D (`com.sun.j3d.utils.geometry.Sphere`) al quale è stata data una colorazione e delle luci per risaltare l'effetto tridimensionale. Inoltre per rendere più chiara la posizione dell'oggetto all'interno dello spazio tridimensionale sono stati aggiunti gli assi cartesiani x e y attraverso la creazione di due linee 3D. La posizione della sfera nello spazio 3D viene catturata mediante il metodo `get2dCoordinate()` che restituisce un vettore delle coordinate. Spetterà poi all'agente che controlla questo sensore ricavare l'accelerazione relativa allo spostamento effettuato dalla sfera nello spazio. L'interazione con la sfera, per simulare eventuali movimenti della persona sotto monitoraggio, è svolta aggiungendo al gruppo di cui fa parte l'oggetto 3D un mouse behaviour di tipo traslatorio⁴.

L'interfaccia nel caso dei sensori quali termometro, sfigmomanometro, cardiofrequenzimetro è composta principalmente da un pannello diviso in due parti: nella prima vi è il nome della misura, nella seconda un valore modificabile attraverso le apposite frecce. La modifica sui valori cambia a seconda del sensore: nel caso della temperatura, il passo di cambiamento è di 0,1; nel caso della pressione è di 1, così come nel caso della frequenza cardiaca. Nell'interfaccia dello sfigmomanometro, infine, il pannello è dotato di due righe identiche, una per la pressione minima, l'altra per la massima. Gli agenti che "dialogano" con questi sensori si limitano semplicemente a invocare un metodo pubblico che restituisce il valore corrente sull'interfaccia.

4.2 APPLICAZIONE DEL PAZIENTE

L'applicazione del lato paziente si compone di una semplice interfaccia e dall'agente paziente che svolge tutte le attività di interesse del suo utente. I dati dei sensori visualizzati sono:

- L'accelerazione lungo l'asse x
- L'accelerazione lungo l'asse y

⁴ `com.sun.j3d.utils.behaviors.mouse.MouseBehavior`

- La temperatura
- La frequenza cardiaca
- La pressione minima
- La pressione massima

paziente 1@test	
Valore Accelerometro (x)	0.0
Valore Accelerometro (y)	0.0
Valore Temperatura	36.4
Valore Frequenza cardiaca	60.0
Valore Pressione minima	80
Valore Pressione massima	120

FIGURA: INTERFACCIA PAZIENTE

L'agente paziente ha il compito di aggiornare l'interfaccia con i dati dei sensori verificando che non vi siano situazioni di emergenza, inoltre deve depositare i valori raccolti sul database.

All'avvio l'agente paziente registra se stesso presso il noto servizio di pagine gialle di JADE, in modo tale da consentire la ricerca dello stesso agli agenti di emergenza per l'invio di messaggi come verrà spiegato nel capitolo relativo al funzionamento in caso di emergenza.

In seguito l'agente avvia la propria interfaccia con i seguenti valori di default:

- Accelerometro: accelerazione $x = 0$, accelerazione $y = 0$;
- Sfigmomanometro: pressione massima = 120, pressione minima = 80;
- Cardiosfigmomanometro: frequenza cardiaca = 60;
- Termometro: temperatura = 36,4.

Dopo aver avviato l'interfaccia, l'agente paziente crea, sia per gli agenti che fungono da driver per i sensori, sia per quello che funge da driver per il database, due container nominati rispettivamente con il proprio nome (codice della tessera sanitaria) preceduto dalla dicitura "container sensori" nel caso dei container sensori e "container DB driver" nel caso di quello relativo al driver database. All'interno di quello dei driver sensori vengono subito avviati gli agenti sensori dando loro come argomento l'AID dell'agente.

All'arrivo di un messaggio da parte di uno dei sensori, per prima cosa, viene estratto il contenuto in modo da ottenere dei valori utilizzabili. Questi valori in seguito vengono controllati secondo delle soglie predefinite.

Nel caso in cui un valore sia fuori soglia l'agente entra nello stato di emergenza avviando nel suo stesso container, l'agente che si occupa dell'emergenza dandogli tutti i dati necessari. In questo stato e fintantoché non viene ricevuto un messaggio di annullamento o un messaggio che segnali la fine dell'emergenza, l'agente non rileva altre emergenze.

4.3 APPLICAZIONE DEL MEDICO

Come avviene nell'applicazione del paziente, anche quella del medico si può dividere in due parti: l'interfaccia e l'agente medico.

L'interfaccia dell'applicazione è stata disegnata per essere più completa rispetto a quella del paziente: presenta in alto un pannello contenente i dati anagrafici del paziente quali nome, cognome, indirizzo, data di nascita, numero di telefono e numero della tessera sanitaria. Subito sotto questo pannello è presente uno spazio in cui il medico può inserire una stringa di testo per specificare il tipo di emergenza che ha rilevato. Questa stringa di testo sarà utilizzata per lanciare l'emergenza da parte del medico.

Cognome	Test	Nato il	01/01/1980	Tessera Sanitaria	000000001
Nome	Test	Indirizzo	Via Test 1 Test	Telefono	000000001

Emergenza da segnalare

DATI SENSORI

registrazione valori del 2010-07-30 00:35:21.0

valori accelerometro(x):
media: 0.0
massima: 0.0
minima: 0.0

valori accelerometro(y):
media: 0.0

CARTELLA CLINICA

test

Update Cartella Clinica Emergenza Cambia Paziente Chiudi

FIGURA: INTERFACCIA MEDICO

Per quanto riguarda la cartella clinica e i dati dei sensori, dato che queste parti sono composte da innumerevoli parametri (ad esempio per la cartella clinica possono esserci anamnesi, terapie seguite, allergie, ecc), si è pensato, per semplificare la realizzazione dell'interfaccia, di organizzare il tutto in due JTextArea (javax.swing.JTextArea) contenenti la prima i dati dei sensori in ordine cronologico, la seconda la cartella clinica gestita come un testo.

L'utente non può modificare né i dati dei sensori né i dati anagrafici dei pazienti, l'unica parte modificabile è l'area di testo della cartella clinica per l'eventuale aggiornamento sul server.

Infine subito sotto questo pannello è presente quello con dei pulsanti per l'interazione con il medico. Per primo vi è quello per eseguire l'aggiornamento della cartella clinica nel caso in cui vi sia- no state fatte delle modifiche, poi quello per lanciare un'emergenza (e in questo caso va compilato l'apposito campo), infine gli ultimi due servono a cambiare paziente che si vuole consultare e a chiudere e disconnettere dal sistema il medico.

Come detto precedentemente, l'applicazione del medico è gestito dall'agente medico. Questo, come accade nel caso di quello del paziente viene lanciato da uno starter che si occupa di dare un nome univoco all'agente e al container che lo contiene (nella realizzazione si è pensato di utilizzare per entrambi il codice personale del medico). Una volta lanciato, per prima cosa, questo agente si registra presso il DF per consentire l'eventuale ricerca dell'agente medico da parte di altri agenti e in seguito lancia anch'egli un agente driver in un container separato per comunicare al database che il medico è attivo nel sistema.

4.4 APPLICAZIONE DELL'OPERATORE SANITARIO SUL TERRITORIO

L'applicazione realizzata per gli OST è molto semplice: dato che la loro attività è quella di entrare in azione solo nel caso in cui si verifichi un'emergenza, si è pensato di trasferire la parte strettamente relativa alla conoscenza e alla gestione dei dati dei pazienti nell'applicazione che migra sull'host dell'OST nel caso venga verificato un evento critico.

La funzione principale dell'applicazione dell'OST è quindi solo quella di mantenere attivo l'operatore nel sistema per intervenire in caso di emergenza. L'interfaccia sviluppata è composta da un unico pannello contenete un pulsante che consente la disconnessione dal sistema.

L'agente che si occupa di svolgere le funzioni dell'OST, come avviene negli altri casi è avviato da un agente starter che si occupa di collocarlo all'interno di un container. Sia al container, sia all'agente viene dato un nome unico e in questa realizzazione esso è il codice personale dell'OST.

L'azione dell'agente è sequenziale, non vi è gestione di messaggi, non utilizza altri agenti come driver e rimane attivo fintantoché l'OST è attivo nel sistema. Dato che è un agente leggero si è pensato di dotarlo dei protocolli di comunicazione con il database, pertanto all'avvio esegue una connessione con la base di dati aggiornando lo status dell'OST. Questa procedura viene eseguita anche quando l'OST si disconnette dal sistema premendo l'apposito pulsante. Per comunicare con l'interfaccia e gestire il pulsante, l'agente è stato realizzato come un GUI Agent.

4.5 EMERGENZE

La gestione delle emergenze nel sistema sviluppato è affidata a due agenti: l'agente emergenza paziente e l'agente emergenza medico. Entrambi possiedono caratteristiche e interfacce simili. Questi due agenti sono diversi da quelli precedentemente presentati: devono essere in grado di interagire con il medico o con il personale OST e devono essere anche in grado di spostarsi sui loro dispositivi.

Pertanto non sono solo GUI Agent, ma anche Mobile Agent, per far migrare l'agente su una certa posizione. Le interfacce gestite da entrambi gli agenti sono due: la prima è una sorta di finestra di dialogo per chiedere una conferma sull'emergenza, la seconda è un'interfaccia simile a quella del medico sia per l'agente emergenza paziente sia per quello medico. Questo perché dal lato medico l'emergenza ricevuta va segnalata con dati simili a quelli con cui lavora il medico (dati anagrafici del paziente, dati dei sensori e cartella clinica), dal dato OST l'utilizzo dell'applicazione è simile a quella che farebbe il medico.

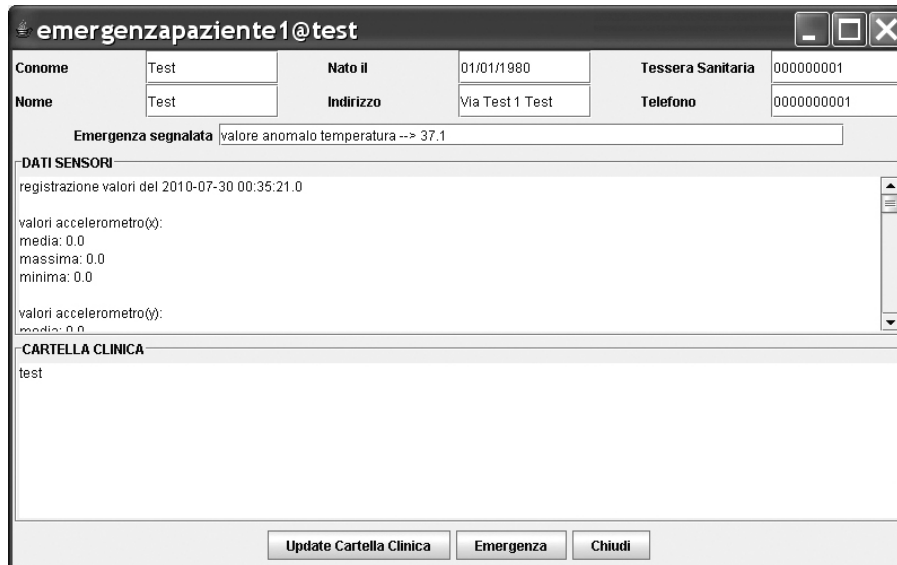


FIGURA: INTERFACCE DELL'AGENTE EMERGENZA PAZIENTE.

Al lato paziente, come già precedentemente detto, l'emergenza scatta non appena il valore rilevato è superiore o inferiore a una soglia fissata avviando l'agente emergenza paziente. Nel caso del medico invece l'emergenza viene attivata premendo l'apposito pulsante avviando anche in questo caso l'agente addetto a segnalare la situazione critica.

Entrambi gli agenti vengono avviati nello stesso container dell'agente che li ha creati. All'avvio inoltre ricevono in argomento una serie di dati fra cui l'emergenza rilevata e l'AID dell'agente che li ha creati. In seguito entrambi lanciano la finestra di dialogo per confermare o meno l'emergenza, se questa non viene confermata semplicemente muoiono e, nel caso dell'agente emergenza paziente, questo comunica al paziente di tornare in uno stato di funzionamento normale. Nel caso in cui l'emergenza venga confermata, per prima cosa entrambi cercano sul database la loro figura professionale di destinazione (il medico per l'agente emergenza paziente, l'OST per l'emergenza medica), una volta trovata, utilizzando il codice di ricerca su DF e AMS della location presentato nel capitolo relativo a JADE, migrano sul container dell'agente trovato. Una volta arrivati, visualizzano l'interfaccia più complessa, avviando il driver database medico per ottenere i dati da visualizzare.

Inoltre entrambi gli agenti hanno la possibilità di effettuare delle modifiche alla cartella clinica. Sia per l'emergenza paziente, sia per l'emergenza medica, queste dovrebbero rappresentare la compi-

di moduli ed eventuali prescrizioni di medicinali o terapie al paziente per risolvere la situazione che ha provocato l'emergenza.

Nell'implementazione si è scelto di non far eseguire più di una migrazione ad un agente: in questo caso se la situazione critica evidenziata dall'agente di emergenza paziente viene confermata dal medico premendo l'apposito tasto di emergenza, questo non migra sulla posizione dell'OST, ma si limita a lanciare l'agente emergenza medico.

4.6 GUASTI E RIDONDANZE

In un sistema comprendente dispositivi eterogenei come quello realizzato, sono possibili dei guasti o malfunzionamenti che potrebbero pregiudicare il funzionamento del sistema stesso.

In fase di progettazione si era pensato che, per ogni dispositivo come quelli dei pazienti, quelli dei medici, il database e i sensori, potessero avere due tipi di malfunzionamento: o smettere di funzionare o funzionare in modo anomalo. Nel primo caso la rilevazione del malfunzionamento sarebbe stata abbastanza semplice implementando un meccanismo di feedback nella comunicazione in modo da accorgersi in tempi brevi se il dispositivo non fosse più funzionante. Nel secondo caso la rilevazione della anomalia sarebbe stata più complicata dato che avrebbe richiesto un'analisi dei dati più approfondita per verificare eventuali valori fuori dal funzionamento ordinario: ad esempio se un sensore continua a trasmettere dati molto variabili fra un campionamento e un altro in un lungo periodo di monitoraggio, questo potrebbe essere indice di un qualche malfunzionamento.

Quando un dispositivo viene meno per guasto è importante avere un meccanismo di ripristino del sistema in modo da poter riportare il tutto ad una situazione ottimale.

Dal punto di vista del paziente si è implementata una procedura di backup che prevede la memorizzazione degli ultimi cinque messaggi spediti al database, questo perché sul dispositivo del paziente ci sono delle restrizioni di spazio dovute alla carenza di memoria poiché si ha a che fare con un apparecchio mobile. Sul lato medico e nelle emergenze, invece vengono prevalentemente usate macchine con memorie decisamente più capienti, quindi si è pensato di eseguire un backup di tutti i dati inviati al database riguardanti la cartella clinica, per il cui aggiornamento viene svolta una visita in scrittura.

5. CONCLUSIONI

L'obiettivo del documento è stato quello di mostrare le caratteristiche tipiche dei MAS e come possano migliorare i sistemi sanitari e assistenziali in termini di interoperabilità, affidabilità, modularità e robustezza e come il personale medico (e, di conseguenza, i cittadini) possa beneficiare di questo sistema distribuito ed efficiente.

Un'architettura distribuita, in questo tipo di scenario, risulta molto utile per gestire questi sistemi. Oltretutto questo paradigma risulta molto attuale e in continua evoluzione fornendo molti spunti di sviluppo, poiché consente da un lato di sfruttare le diverse caratteristiche dei dispositivi che compongono il sistema, dall'altro, con l'introduzione di architetture middleware, consente di affrontare problematiche come eterogeneità (dei dispositivi e dei programmi), guasti, perdita di informazioni, scalabilità e sicurezza in maniera agile ed efficiente.

JADE, realizzato in Java, in particolare, per la natura indipendente e autonoma degli agenti è particolarmente adatto allo sviluppo di un sistema pervasivo, anche grazie al fatto che offre molte libertà al programmatore nella gestione dei comportamenti degli agenti, fornendo un linguaggio semplice e di immediata comprensione, ma non per questo povero di funzionalità.

Il sistema illustrato come case study è solo un prototipo di quale può essere un'applicazione per l'assistenza sanitaria domestica. Questo documento aveva lo scopo di mostrare come si potrebbe realizzare questo tipo di sistemi in questo contesto applicativo.

RIFERIMENTI

- [1] C. Doukas, and T. Pliakas, and I. Maglogiannis, "Mobile healthcare information management utilizing Cloud Computing and Android OS", Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pp. 1037-1040, 2010.
- [2] C. Rolim, F. Koch, C. Westphall, J. Werner, A. Fracalossi, and G. Salvador, "A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions", 2010 Second International Conference on eHealth, Telemedicine, and Social Medicine, pp. 95-99, 2010.
- [3] V. Koufi, F. Malamateniou, Flora and G. Vassilacopoulos, "Building Interoperable Health Information Systems Using Agent and Workflow Technologies", Medical Informatics in a United and Healthy Europe, pp. 180-184, 2009.
- [4] Robert H Dennard, VL Rideout, E Bassous, and AR Leblanc. Design of ion implanted mosfet's with very small physical dimensions. Solid-State Circuits, IEEE Journal of, 9(5):256–268, 1974
- [5] Marcello Coppola, Babak Falsafi, John Goodacre, and George Kornaros. From embedded multi-core socs to scale-out processors. In Proceedings of the Conference on Design, Automation and Test in Europe, pages 947–951. EDA Consortium, 2013
- [6] Erik Dahlman, Stefan Parkvall, and Johan Skold. 4G: LTE/LTE-advanced for mobile broadband. Academic press, 2013
- [7] Bernhard Schulz. Lte transmission modes and beamforming. Rohde and Schwarz White Paper, 2011
- [8] Eldad Perahia and Michelle X Gong. Gigabit wireless lans: an overview of ieee 802.11 ac and 802.11 ad. ACM SIGMOBILE Mobile Computing and Communications Review, 15(3):23–33, 2011
- [9] Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. Computer communications, 30(7):1655–1695, 2007
- [10] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. Sensors, 12(9):11734–11753, 2012
- [11] K. Ashton, "That Internet of Things thing," RfID Journal, vol. 22, no. 7, pp. 97–114, 2009
- [12] Roy Want. An introduction to rfid technology. Pervasive Computing, IEEE, 5(1):25–33, 2006
- [13] Edmund W Schuster, Stuart J Allen, and David L Brock. Global RFID: the value of the EPCglobal network for supply chain management. Springer Science & Business Media, 2007
- [14] Loic Schmidt, Nathalie Mitton, and David Simplot-Ryl. Towards unified tag data translation for the internet of things. In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology. Wireless VITAE 2009, pages 332–335. IEEE, 2009
- [15] Chonggang Wang, Tao Jiang, and Qian Zhang. ZigBee R network protocols and applications. CRC Press, 2014

- [16] Lui Sha, Tarek Abdelzaher, Karl-Erik Arzen, Anton Cervin, Theodore Baker, Alan Burns, Giorgio Buttazzo, Marco Caccamo, John Lehoczky, and Aloysius K Mok. Real time scheduling theory: A historical perspective. *Real-time systems*, 28(2-3):101–155, 2004
- [17] Joseph M. Kahn, Randy Howard Katz, Kristofer S. J. Pister. Mobile networking for smart dust. In *ACM/IEEE Int. Conference on Mobile Computing and Networking (MobiCom 99)*, Seattle, 1999
- [18] Roberto Verdone, Davide Dardari, Gianluca Mazzini, and Andrea Conti. *Wireless sensor and actuator networks: technologies, analysis and design*. Academic Press, 2010
- [19] Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007
- [20] CL Philip Chen and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347, 2014
- [21] Jay Lee, Edzel Lapira, Behrad Bagheri, and Hung-an Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, 1(1):38–41, 2013
- [22] D. J. Cook, S. K. Das, “How smart are our environments? An updated look at the state of the art”, *Pervasive Mob. Comput.*, vol. 3, no. 2, pp. 53-73, 2007
- [23] C. Badica, M. Brezovan, A. Badica. “An Overview of Smart Home Environments: Architectures, Technologies and Applications”, in *Proceedings of the Sixth Balkan Conference in Informatics*, ser. BCI 2013, C. K. Georgiadis, P. Kefalas, and D. Stamatias, Eds., vol. 1036. CEUR-WS.org, 2013
- [24] A. Omicini and F. Zambonelli. “Challenges and research directions in agent-oriented software engineering”, *Autonomous Agents and Multi-Agent Systems*, 9(3):253–283, November 2004.
- [25] M. Luck, P. McBurney, and C. Preist. “A manifesto for agent technology: Towards next generation computing”, *Autonomous Agents and Multi-Agent Systems*, 9(3):203–252, 2004.
- [26] M. Wooldridge and N.R. “Jennings. *Intelligent agents: Theory and practice*”, *Knowledge Engineering Review*, 10(1):115–152, 1995.
- [27] R. Levy. “Representing agent and their system: A challenge for current modelling”, *Informatica*, 28(1):3–11, 2004.
- [28] A. Nguyen, T. Nakano, and T. Suda. “Learning from nature: Network architecture inspired by biology”, *ACM Crossroads*, 11(4):3–7, 2005.
- [29] D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran¹, and J. White. “MASIF: The OMG Mobile Agent System Interoperability Facility”, in *Mobility: processes, computers, and agents*, pages 628–641. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999.
- [30] FIPA. <http://www.fipa.org>.