

Ingineria software a sistemelor ambientale inteligente

Manate Bogdan, master IS, an 2

30 ianuarie 2011

Abstract

Inteligența ambientală implică sintetizarea datelor dintr-o largă gamă de surse pentru a prezenta un comportament adaptiv relevant, fără implicarea directă a utilizatorului. Sintetizarea se face pe date provenite de la dispozitive de măsură independente sau alte surse de date. În zilele noastre sistemele inteligente sunt omniprezente în toate domeniile, fiind prezente chiar și în viața noastră de zi cu zi. Mediile inteligente se pot integra peste tot, de la case de locuit prevăzute cu dispozitive automate de control, la birouri sau hoteluri care au din ce în ce mai multe dispozitive inteligente pentru a asista muncitorii în activitățile pe care le desfășoară, pentru a economisii energie sau pentru a satisface preferințele clienților.

1 Introducere

Putem defini un mediu inteligent ca unul care este capabil să dobândească și să aplice cunoștințele despre mediu și locuitorii săi, în scopul de a îmbunătăți experiența lor în acest mediu.

Sistemele ambientale inteligente se pot clasifica în două categorii: sisteme contruite din rețele independente și dispozitive programabile, care au funcționalități restrânse și sisteme care trebuie să producă un comportament centrat pe utilizator și care trebuie să reacționeze la scenarii complexe într-un mod care păstrează un accent clar pe activitățile desfășurate de utilizatori zi de zi. Deși suntem atrași să credem că un comportament centrat pe utilizator va reieși din interacțiunea dintre diverse dispozitive, există motive puternice care sugerează o abordare mai directă în cele mai multe cazuri. În timp ce numeroase aplicații la scară mică și-au dovedit implică utilitatea, aplicațiile mari rămân încă evazive, deoarece este foarte greu să se anticipeze ce se întâmplă în lumea reală și cu atât mai puțin să se traducă aceste lucruri într-un comportament adaptiv relevant, de asemenea este foarte greu să se administreze greșelile inevitabile care intervin în timpo viziune de ansamblu a răspunsurilor sistemului la mediul său în schimbare ce se execută acțiuni complexe, fără intervenția utilizatorului.

În general sistemele ambientale inteligente ridică următoarele întrebări:

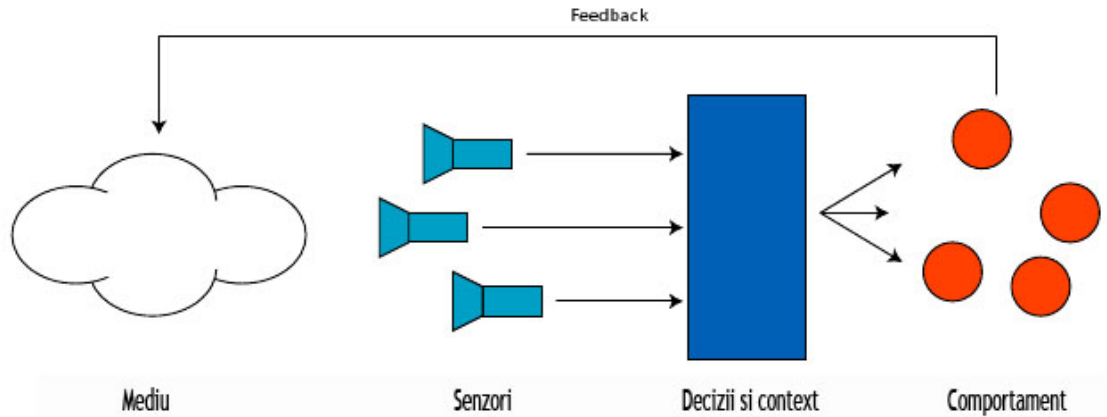


Figure 1: Module de mapare a datelor provenite de la senzori pe un anumit comportament

1. Ce impact are calitatea scăzută, natura difuză și eterogenă a datelor provenite de la senzori și operația de inferență în programare ?
2. Ce implicații au atributele de la punctul 1 în crearea de software middleware sau cadre de programare pentru inteligența ambientală ?

Sistemele ambientale inteligente nu se bazează doar pe detectarea stimulilor interni sau externi mai degrabă se bazează pe valorificarea informațiilor dintr-o gamă diversă de surse interconectate. Acest lucru schimbă percepția de abordare, care pare a fi o problemă de detectare și de raționament la una de management al informației și mentenanță: o viziune de ansamblu a răspunsurilor sistemului la mediul său în schimbare.

2 Incertitudinea datelor provenite de la senzori

Aplicațiile tradiționale oferă un anumit context în cadrul căruia pot să apară diferite interacțiuni. Pentru majoritatea sistemelor există un grad scăzut de incertitudine în cadrul acestor interacțiuni. Un utilizator poate selecta sau nu un meniu sau apăsa o anumită tastă. În cadrul anumitor sisteme pot exista erori datorită mesajelor pierdute, care conțin erori sau care sunt trimise repetitiv, aceste erori fiind tratate în majoritatea cazurilor în cadrul middlewareului. În majoritatea sistemelor există de asemenea erori de operare care trebuie corectate într-un fel sau altul, de obicei cu un impact minim asupra sistemului. Cu toate că unele interacțiuni pot genera erori în cadrul acestor sisteme, aceste erori sunt definite în prealabil.

Interacțiunile ambientale prezintă caracteristici complet diferite. Interacțiunile apar, în cea mai mare parte, în urma observării de către senzori a acțiunilor

de zi cu zi. Chiar și în cadrul unei singure categorii de senzori, cum ar fi cei de locație, fiecare categorie de senzori are de cele mai multe ori caracteristici radical diferite. De exemplu, un sistem Wifi de localizare, cum ar fi PlaceLab, poate raporta în general poziția unui dispozitiv cu precizie de 10 m, dar s-ar putea din cauza reflexie ca senzorul să raporteze o locație diferită față de locația fizică în care se află dispozitivul. Aceste condiții se pot aplica analog și la celelalte tipuri de senzori. O aplicație proiectată după modelul din Figura 1 ar putea să execute următoare instrucțiune: “Când A și B sunt în camera X execută acțiunea 1”, nici un senzor nu poate să ofere informații precise despre locație pentru a fi luată o decizie cu încredere.

Problema de bază este că nu putem atribui instrucțiuni la o singură acțiune, ci doar la un anumit număr de acțiuni întreprinse în context.

3 Detectare și acțiune

Sistemele orientate pe evenimente rezolvă această problemă prin oferirea unui model de sarcini care este implementat de aplicație. Evenimentele primite sunt folosite pentru a executa diferite sarcini. Lucrurile devin și mai complicate când luăm în considerare sistemele ambientale care conțin mai multe aplicații sau servicii. O anumită acțiune poate fi interpretată de mai multe aplicații simultan, fiecare dintre care poate prezenta apoi un comportament. Aceste comportamente sunt independente, cu toate acestea comportamentele pot fi și dependente, un anumit comportament fiind afectat negativ de alt comportament al aplicației.

4 Proiectarea și programarea sistemelor având în vedere incertitudinea datelor

Toate datele folosite într-un sistem ambiental inteligent pot fi considerate nesigure, acestea ridicând probleme pe care tehnologia actuală de construire a senzorilor nu le poate rezolva. Relațiile dintr datele provenite de la senzori și scenariile suportate de sistemele ambientale nu pot fi definite exact, deoarece de cele mai multe ori unei acțiuni nu îi poate fi atribuit fără echivoc, un rol particular într-un proces care face obiectul incertitudinilor acțiunilor de zi cu zi ale utilizatorilor. Acest lucru implică faptul că orice decizie luată în timpul execuției unei sarcini poate fi greșită, în măsura în care datele au fost greșit interpretate.

Cu toate că senzorii sunt imprecisi individual, un mediu care dispune de un număr mare de senzori poate compensa aceste deficiențe individuale și poate realiza o estimare apropiată de valoarea reală sau chiar exactă. Acest lucru poate ridica o problemă legată de numărul de senzori, existenți într-un mediu.

Sistemele ambientale pot îmbunătăți estimările datelor, folosindu-se de date citite în trecut, de valorile implicite și de datele citite de la senzori. Un punct important care trebuie luat în considerare este decuplarea comportamentului de

observațiile făcute izolat și de a folosi în schimb un model global al mediului, care poate fi determinat de toate aspectele mediului care pot fi detectate de senzori. Pentru a evita ca datele eronate să fie propagate în sistem, trebuie luate în considerare următoarele:

- Evenimentele pot propaga erori, drept urmare nu pot servi ca bază pentru programare. Un eveniment provenit de la un senzor, cum ar fi poziția unui individ într-un anumit spațiu, poate fi eronat. Aplicația va reacționa la acest eveniment declanșat de date eronate, lucru care nu se dorește, drept urmare aplicația trebuie să identifice datele eronate local, înainte de a executa acțiuni.
- Evitarea acțiunilor asupra datelor provenite de la un singur senzor. Pentru a evita datele eronate, acțiunile trebuie luate pe baza datelor preluate de la mai multe surse.
- Interconectarea este mai importantă decât datele. Nu putem considera un sistem ambiental inteligent în cazul în care este pur și simplu condus de evenimente provenite de la senzori, neținând cont și de alți factori. Modelul, relațiile sale și capacitatea de a extrage informații prin intermediul fuziunii caracterizează sistemul mai mult decât disponibilitatea senzorilor.
- Orice decizie are nevoie de o strategie de atenuare. Este inevitabilă apariția erorilor, iar aceste erori pot cauza probleme majore pentru aplicație dacă nu sunt luate în calcul la proiectarea sistemului.
- Lucrurile interesante vin din compoziție. Compunerea aplicațiilor și serviciilor au impact asupra întregului sistem, de aceea rezultatele compoziției trebuie să fie definite din timp.

În proiectarea unui sistem ambiental inteligent ar trebui să se țină cont de următoarele procese: percepție, raționament și acțiune. Aceste trei procese trebuie să interacționeze continuu pentru a realiza un sistem ambiental complet.

5 Percepția

Procesul de percepție ar trebui să fie împărțit în diferite sarcini, în scopul de a oferi o percepție corectă asupra lumii reale. Acesta trebuie să se ocupe cu detalii low-level pentru a prelua datele de la lumea reală și să le adapteze la o bază de cunoștințe. Acest proces trebuie să șteargă datele eronate, nesemnificative sau valorile redundante în scopul de a obține acuratețea necesară proceselor următoare. Procesul de percepție este împărțit în 5 etape: colectarea datelor, verificarea datelor, corectarea datelor, filtrarea datelor și ontologizarea datelor.

5.1 Colectarea datelor

Colectarea datelor este o sarcină de nivel scăzut, obiectivul său principal fiind colectarea datelor de la dispozitivele fizice. Această sarcină va interacționa în

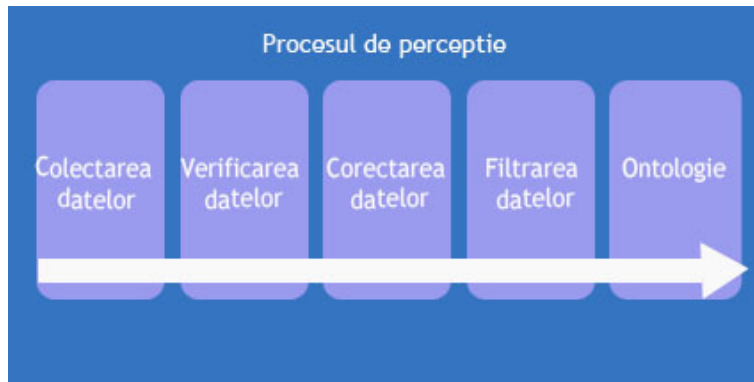


Figure 2: Procesul de percepție

general cu toate dispozitivele plasate în mediu monitorizat de sistemului ambiental inteligent. Pentru ca sarcina de colectare a datelor să ruleze cât mai eficient senzorii trebuie amplasați cât mai organizat pentru a se evita informația redundantă sau nesemnificativă și pentru a reduce costurile. De asemenea trebuie luat în calcul și tipurile de dispozitive folosite, acestea putând fi programabile sau nu. Datorită faptului că majoritatea dispozitivelor fiind programate de către producător, sarcina de colectare a datelor trebuie să fie realizată ținând cont de diferitele implementări ale dispozitivelor.

5.2 Verificarea datelor

Scopul principal al acestei sarcini este de a verifica datele colectate în etapa de colectare a datelor. În această etapă se poate verifica dacă datele sunt corecte sau nu. Adevărata provocare din parte programatorului este să verifice corectitudinea datelor. Datorită faptului că nu există o soluție universală pentru toate mediile existente, sarcina de verificare trebuie să se adapteze în mod dinamic la fiecare mediu.

În cadru acestei sarcini se poate introduce un motor de reguli, care poate rula reguli de verificare pentru a determina dacă datele sunt corecte sau nu pentru mediul curent. Motorul de reguli trebuie să le ofere programatorilor o modalitate flexibilă de a adăuga, modifica sau șterge reguli.

Această sarcină trebuie să ruleze în strânsă legătură cu sarcina de verificare a datelor. Sarcina de verificare a datelor trebuie să poată invoca în mod direct sarcina de corectare.

Verificarea datelor poate fi privită ca un filtru aplicat asupra datelor primite care poate fi folosit să accepte date sau să le respingă din diferite motive (incorecte, redundante sau alte motive adăugate de programatori.

5.3 Corectarea datelor

Scopul acestei sarcini este de a corecta datele incorecte raportate în etapa de verificare a datelor. Datele pot fi corectate în diferite moduri:

- Ignorarea datelor - acest lucru presupune setarea unei valori implicite (ex: valoarea null pentru SQL).
- Ajustarea datelor - se poate modifica datele atribuindu-le ultima valoare corectă înregistrată.
- Respingerea datelor - dacă nu se poate aplica nici una dintre metodele de mai sus datele pot fi respinse.

5.4 Ontologizarea datelor

Algoritmi de inteligență artificială au nevoie de o bază solidă de cunoștințe pentru a funcționa. Scopul acestei sarcini este acela de a organiza datele conform unui model din lumea reală. Această sarcină va trebui să sincronizeze toate datele care provin dintr-un mediu populat cu dispozitive și evenimente asincrone. Sistemele ambientale inteligente au nevoie uneori de seturi de date compuse din valorile citite de la mai multe dispozitive. Implementarea unui proces de sincronizare poate varia, dar majoritatea sistemelor inteligente au în comun următoarele elemente:

- Un buffer de date.
- Un garbage collector.

După ce datele sunt agregate acestea pot fi stocate într-o bază de cunoștințe care va fi accesată de sarcina care se ocupă de raționament sau chiar sarcina care se ocupă de procesul de învățare.

6 Raționamentul

În cadrul sistemelor inteligente procesul de raționament poate fi împărțit în trei sarcini distincte: învățare, raționament și predicție. Procesul de învățare poate fi privit ca un liant pentru celelalte sarcini care au ca scop procesul de raționament. Pentru realizarea procesului de învățare se pot folosi două familii de algoritmi specifici pentru sistemele inteligente care integrează acest proces. Cele două familii de algoritmi sunt SVM (support vector machine) și NN (neural networks). Cel mai mare avantaj al acestor algoritmi este că aceștia returnează tot timpul date la sfârșitul execuției, pe de altă parte aceste date sunt greu de înțeles. Procesul de învățare se poate baza de asemenea pe data mining, proces care se ocupă cu extragerea informațiilor și găsirea anumitor tipare din datele salvate în baza de cunoștințe.

6.1 Recunoașterea situațiilor

O situație este definită ca un set de stări care au valori similare. Un set poate fi definit de:

- o stare canonică reprezentativă pentru toate stările din set.
- o funcție care definește cum se pot măsura diferențele dintre două stări.
- un prag care definește diferența maximă dintre starea canonică și celelalte stări din set.
- o valoare pentru fiecare element al unei stări din intervalul $[0,1]$. Valoarea 0 indicând faptul că starea respectivă poate fi ignorată.

6.2 Predicția situațiilor

Cunoștințele dobândite în urma procesului de învățare sunt utile pentru a face predicții despre următoarele acțiuni care pot fi luate în calcul pentru a îndeplini automat unele obiective. Aceste acțiuni pot fi executate pe baza datelor de ieșire de la algoritmi specifici sistemelor inteligente. Exemplu de predicții:

- Prezicerea evenimentelor și stărilor sistemului.
- Prezicerea acțiunilor locuitorilor mediului.
- Prezicerea anomaliilor.

6.3 Detectarea erorilor

Scopul principal al acestei sarcini este de a supraveghea acțiunile luate de locuitori, pentru a detecta decizii opuse între sistemul ambiental și locuitori. Prin urmare ultimele acțiuni luate de sistemul inteligent trebuie memorate și comparate cu acțiunile luate de locuitori. Compararea acestor acțiuni presupune de obicei operațiuni complexe la care iau parte multe elemente componente ale sistemului și de asemenea trebuie luat în considerare felul în care aceste acțiuni sunt comparate, durata de timp alocată pentru comparare, sau dacă acțiuni asupra diferitelor componente pot fi considerate opuse. Toate aceste aspecte fac detectarea acțiunilor opuse mult mai provocatoare.

7 Acțiuni

Sistemele ambientale inteligente trebuie să acționeze independente pentru a fi considerate sisteme inteligente. Deciziile și sarcinile ordonate de procesul de raționament, trebuie să treacă prin trei sarcini principale ale acestui proces:

Policy manager - decizia executării unei sarcini trebuie supervizată de acest proces. Un sistem inteligent trebuie să definească parametrii pentru economisirea energiei, securitate și confort. Acest proces trebuie să poată modifica parametrii în timpul execuției, deoarece preferințele locuitorilor pot varia.

Planificatorul de sarcini - acest proces trebuie să poată planifica sarcinile în funcție de timpul de execuție sau de prioritate.

Procesul care execută sarcini - acest proces trebuie să interacționeze cu protocoale low level pentru a trimite instrucțiuni dispozitivelor.

References

- [1] Smart Environment Software Reference Architecture, A. Fernandez-Montes, J. A. Ortega, J. A. Alvarez, L. Gonzalez-Abril.
- [2] Whole-System Programming of Adaptive Ambient Intelligence, Simon Dobson, Paddy Nixon.